Ch. 9 – BGP (Part 1)

Cabrillo College

CCNP 1 version 3.0

Rick Graziani

Cabrillo College

Note to instructors

- If you have downloaded this presentation from the Cisco Networking Academy Community FTP Center, this may not be my latest version of this PowerPoint.
- For the latest PowerPoints for all my CCNA, CCNP, and Wireless classes, please go to my web site:

  http://www.cabrillo.edu/~rgraziani/

  - The username is *cisco* and the password is *perlman* for all of my materials.
- If you have any questions on any of my materials or the curriculum, please feel free to email me at graziani@cabrillo.edu (I really don't mind helping.) Also, if you run across any typos or errors in my presentations, please let me know.
- I will add "(Updated – *date*)" next to each presentation on my web site that has been updated since these have been uploaded to the FTP center.

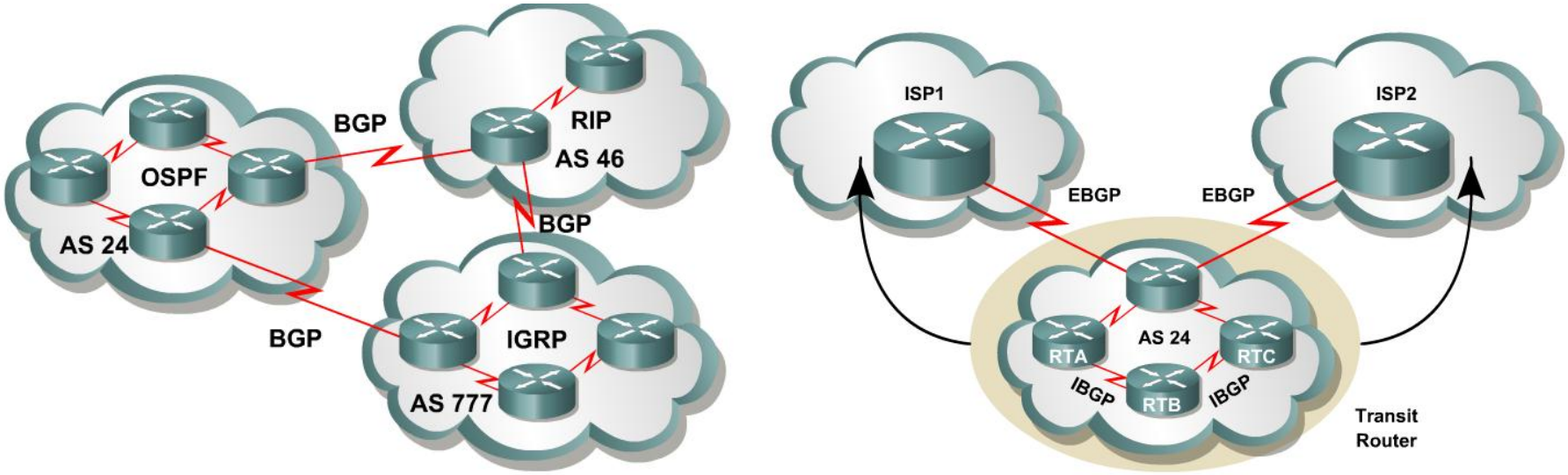
Thanks! Rick

Upon completion of this module, the student will be able to perform tasks related to the following:	
9.1	Autonomous Systems
9.2	Basic BGP Operation
9.3	Configuring BGP
9.4	Monitoring BGP Operation
9.5	The BGP Routing Process
9.6	BGP Attributes
9.7	The BGP Decision Process
9.8	BGP Route Filtering and Policy Routing
9.9	Redundancy, Symmetry, and Load Balancing
9.10	BGP Redistribution
9.11	BGP Configuration Lab Exercises
9.12	Configuring BGP Challenge Lab Exercise

Terms

- **IGP** (Interior Gateway Protocol) - RIP, IGRP, EIGRP, OSPF = Routing protocol used to exchange routing information within an autonomous system.
- **EGP** (Exterior Gateway Protocol) - BGP = Routing protocol used to exchange routing information between autonomous systems.
- **Autonomous System** = (From RFC 1771) “A set of routers under the single technical administration, using an IGP and common metrics to route packets within the AS, and using an EGP to route packets to other AS’s.”
- **BGP** is a path vector or an advanced distance vector routing protocol.

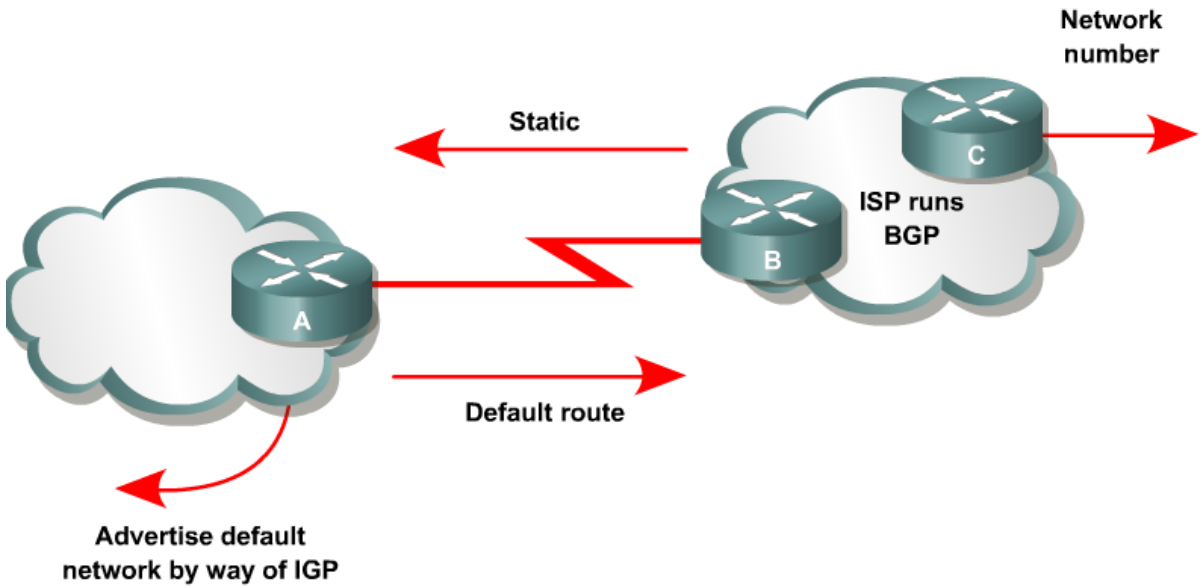
When to use BGP and when not to use BGP – Cisco CCO



Use BGP when the effects of BGP are well understood and one of the following conditions exist:

- The AS allows packets to transit through it to reach another AS (transit AS).
- The AS has multiple connections to other AS's.
- The flow of traffic entering or exiting the AS must be manipulated. This is policy based routing and based on **attributes**.

When to use BGP and when not to use BGP



Do not use BGP if you have one or more of the following conditions:

- A single connection to the Internet or another AS
- No concern for routing policy or routing selection
- A lack of memory or processing power on your routers to handle constant BGP updates
- A limited understanding of route filtering and BGP path selection process
- Low bandwidth between AS's

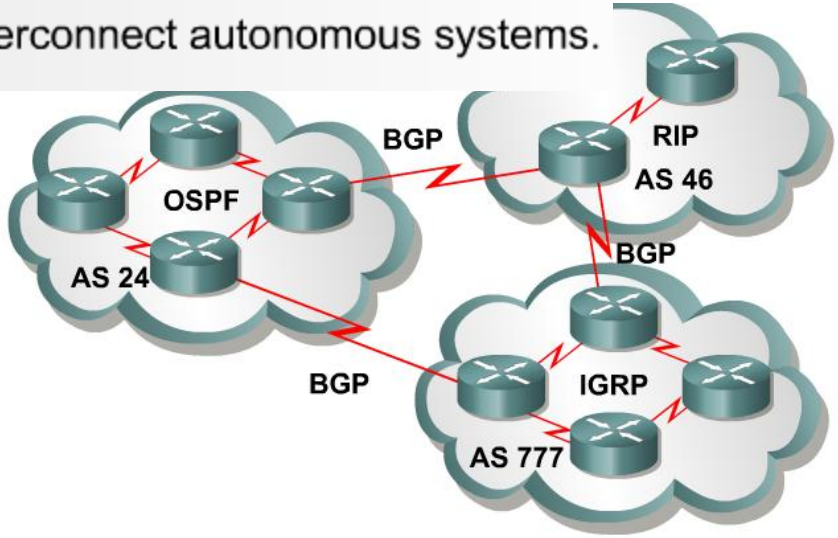
Who needs BGP?

- Not as many internetworks as you may think.
- “You should implement BGP only when a sound engineering reason compels you to do so, such as when the IGPs do not provide the tools necessary to implement the required routing policies or when the size of the routing table cannot be controlled with summarization.”
- “The majority of the cases calling for BGP involve Internet connectivity – either between a subscriber and an ISP or (more likely) between ISPs.”
- “Yet even when interconnecting autonomous systems, BGP might be unnecessary.”

Jeff Dolye, Routing TCP/IP Vol. II

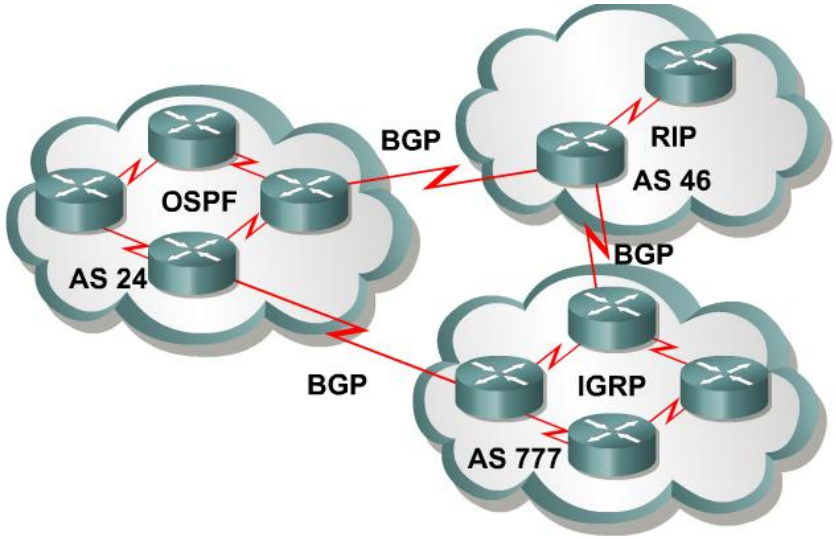
Overview of autonomous systems

EGPs, such as BGP, are used to interconnect autonomous systems.



- An AS is a group of routers that share similar routing policies and operate within a single administrative domain.
- An AS can be a collection of routers running a single IGP, or it can be a collection of routers running different protocols all belonging to one organization.
- In either case, the outside world views the entire Autonomous System as a single entity.

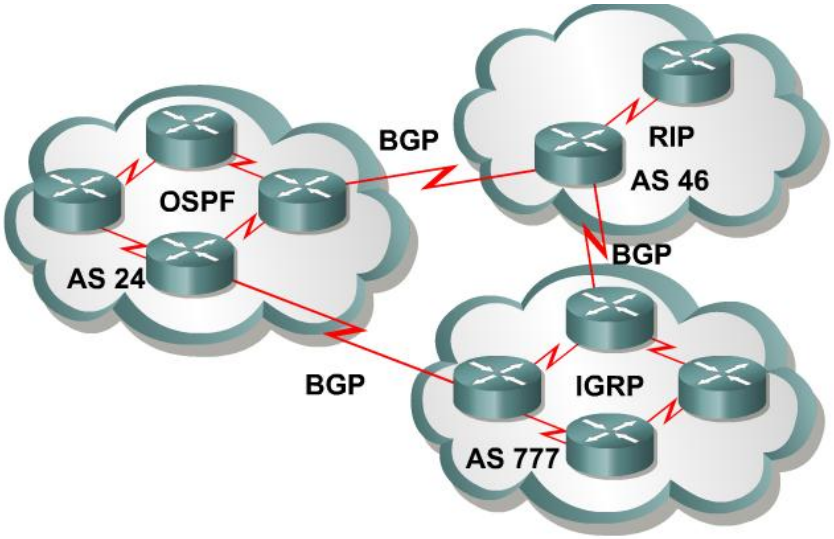
Overview of autonomous systems



AS Numbers

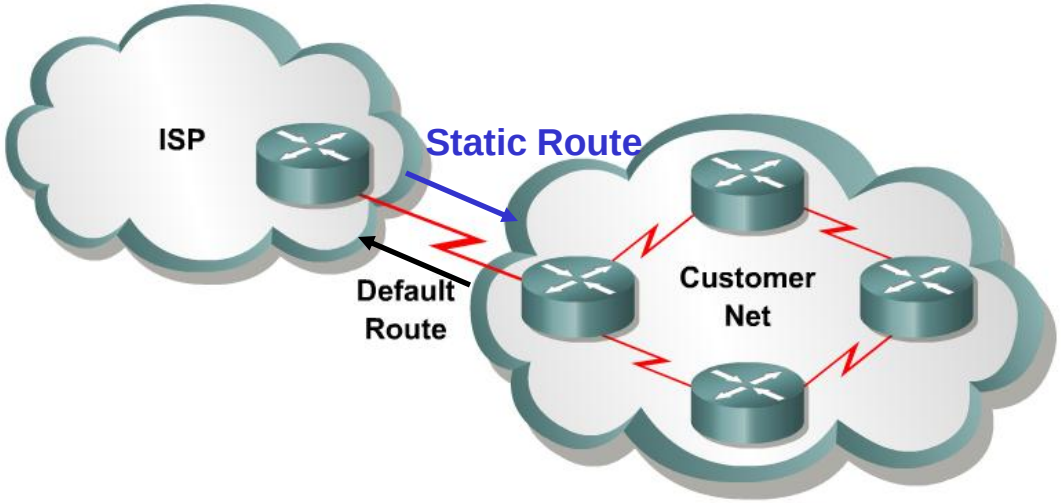
- Each AS has an identifying number that is assigned by an Internet registry or a service provider.
- This number is between **1 and 65,535**.
- AS numbers within the range of **64,512 through 65,535** are reserved for **private** use.
- This is similar to RFC 1918 IP addresses.
- Because of the finite number of available AS numbers, an organization must present justification of its need before it will be assigned an AS number.

Overview of autonomous systems



- Today, the Internet Assigned Numbers Authority (IANA) is enforcing a policy whereby organizations that connect to a single provider and share the provider's routing policies use an AS number from the private pool, 64,512 to 65,535.

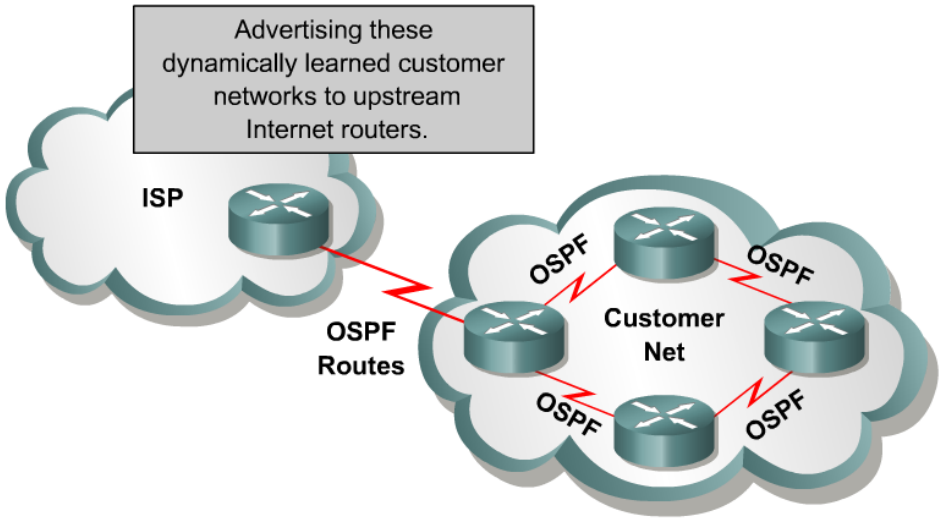
Single-homed autonomous systems



A single-homed AS can be configured with a default route to reach outside networks.

- If an AS has only **one exit point** to outside networks, it is considered a **single-homed system**.
- Single-homed autonomous systems are often referred to as **stub** networks or stubs.
- Stubs can rely on a **default route** to handle all traffic destined for non-local networks.
- BGP is **not** normally needed in this situation.

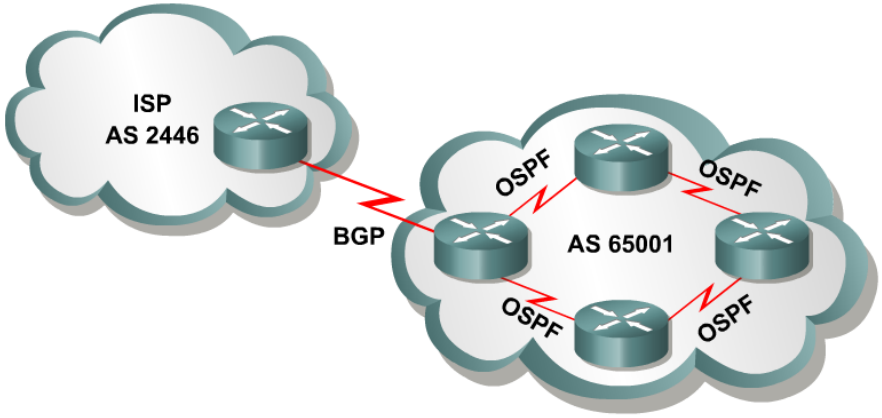
Single-homed autonomous systems



A provider may choose to dynamically learn customer routes using an IGP, such as OSPF.

- Use an IGP – Both the provider and the customer use an **IGP** to share information regarding the customer's networks.
- This provides the benefits associated with dynamic routing.
- BGP is **not** normally needed in this situation.

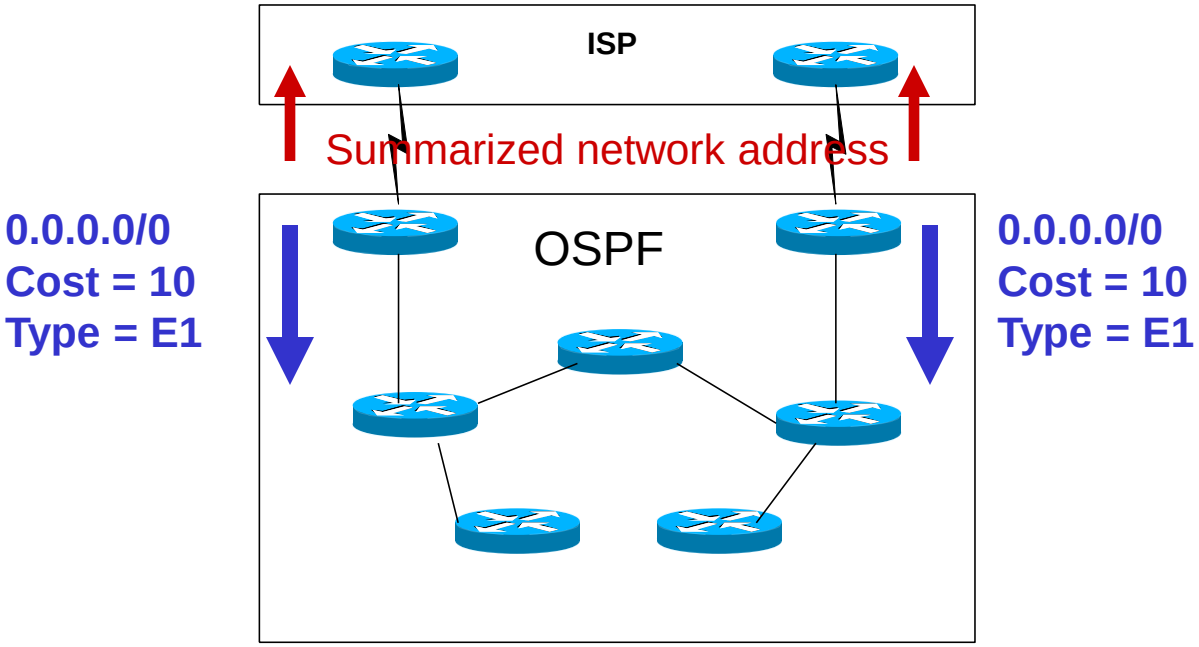
Single-homed autonomous systems



A provider may also choose to dynamically learn a customer's routes using BGP, which typically runs between the ISP router and the customer's boundary router.

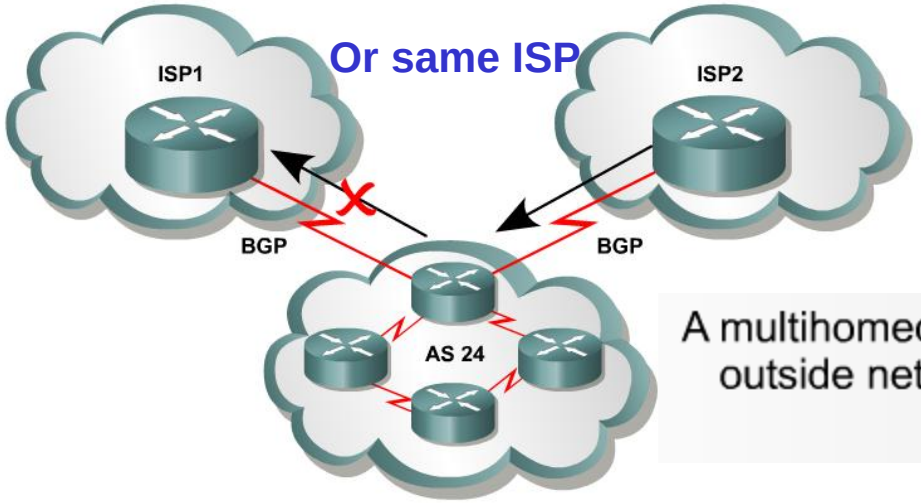
- **Use an EGP** – The third method by which the ISP can learn and advertise the customer's routes is to use an EGP such as BGP.
- In a **single-homed autonomous system** the *customer's routing policies are an extension of the policies of the provider*.
  - For this reason the Internet number registries are unlikely to assign an AS number.
  - Instead, the provider can give the customer an AS number from the **private** pool of AS numbers, 64,512 to 65,535.
  - The provider will strip off these numbers when advertising the customer's routes towards the core of the Internet.

Multi-homed to a Single Autonomous Systems



- This is an improved topology over Single-Home AS, providing for **redundancy**.
- One option may be to use one link as the **primary** link and the other as a **backup** link.
- A better design would be to **use both paths**, with each one providing backup for the other in the event of link or router failure.
- In most cases this will be sufficient for good internetwork performance.

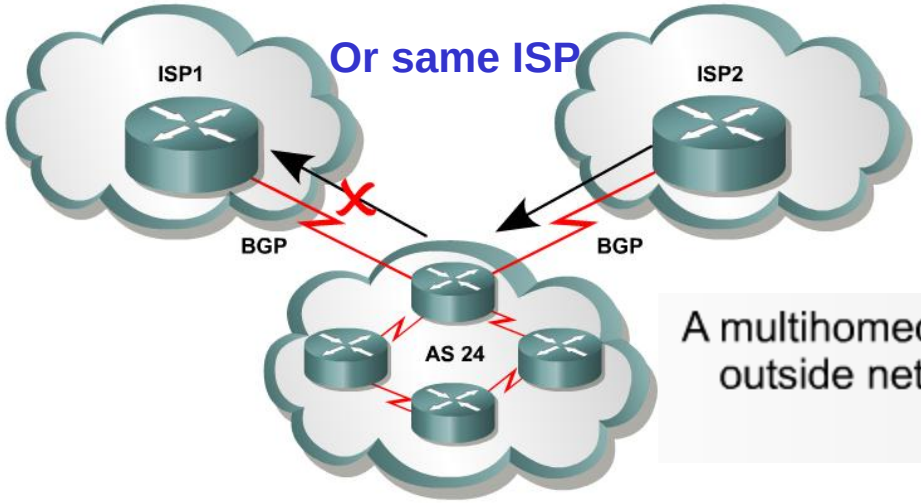
Multihomed nontransit autonomous systems



A multihomed nontransit AS features more than one exit point to outside networks, but does not allow traffic to pass from one outside connection to another.

- An AS is a **multihomed system** if it has *more than one exit point* to outside networks.
- A **non-transit AS** does not allow transit traffic-that is, any traffic that has a source and destination outside the AS—to pass through it.
- A **non-transit AS** would advertise only its *own* routes to both the providers it connects to—it would not advertise routes it learned from one provider to another.
- This makes certain that ISP1 will not use AS 24 to reach destinations that belong to ISP2, and ISP2 would not use AS 24 to reach destinations that belong to ISP1.

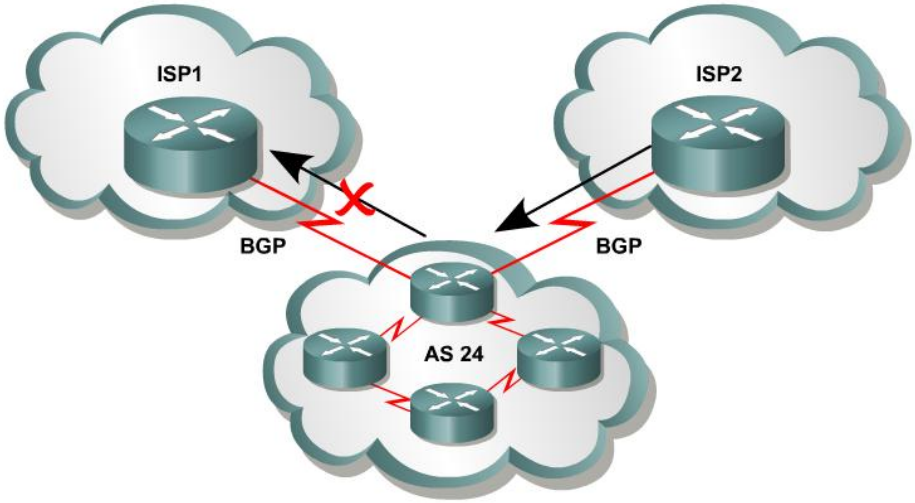
Multihomed nontransit autonomous systems



A multihomed nontransit AS features more than one exit point to outside networks, but does not allow traffic to pass from one outside connection to another.

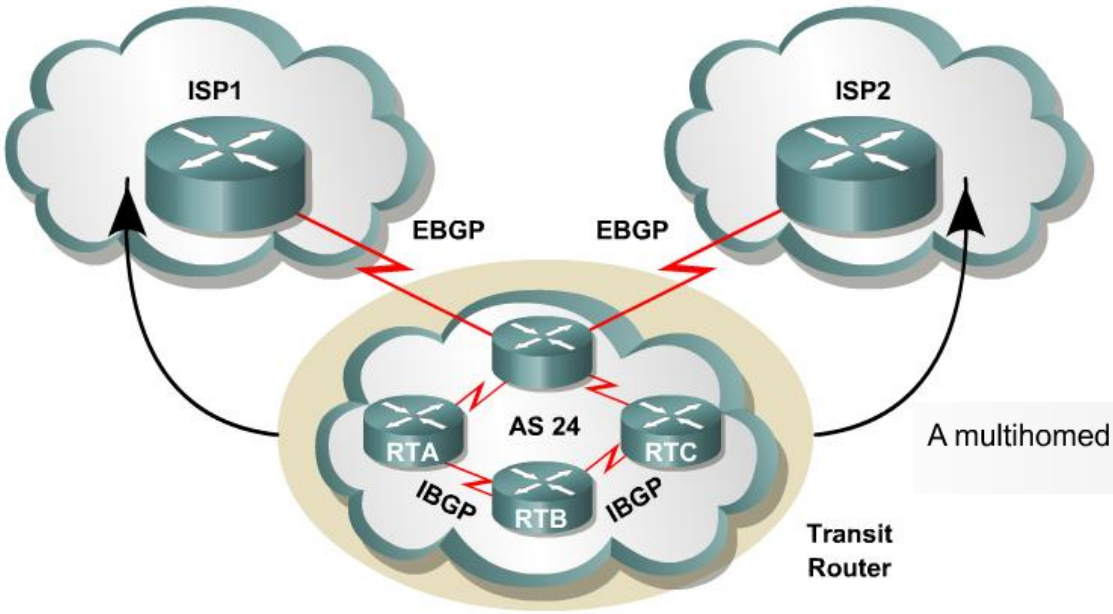
- Multihomed nontransit autonomous systems do not really need to run BGP4 with their providers.
- It is usually **recommended** and **often required** by ISPs.
- As it will be seen later in this module, BGP4 offers numerous advantages, including increased control of route propagation and filtering.

Multihomed nontransit autonomous systems



- ***Incoming route advertisements influence your outgoing traffic, and outgoing advertisements influence your incoming traffic.***
- If the provider advertises routes into your AS via BGP, your internal routers have more accurate information about external destinations.
  - BGP also provides tools for setting routing policies for external destinations.
- If your internal routes are advertised to the provider via BGP, you have influence over which routes are advertised at which exit point.
  - BGP also provides tools for your influencing (to some degree) the choices the provider makes when sending traffic into your AS.

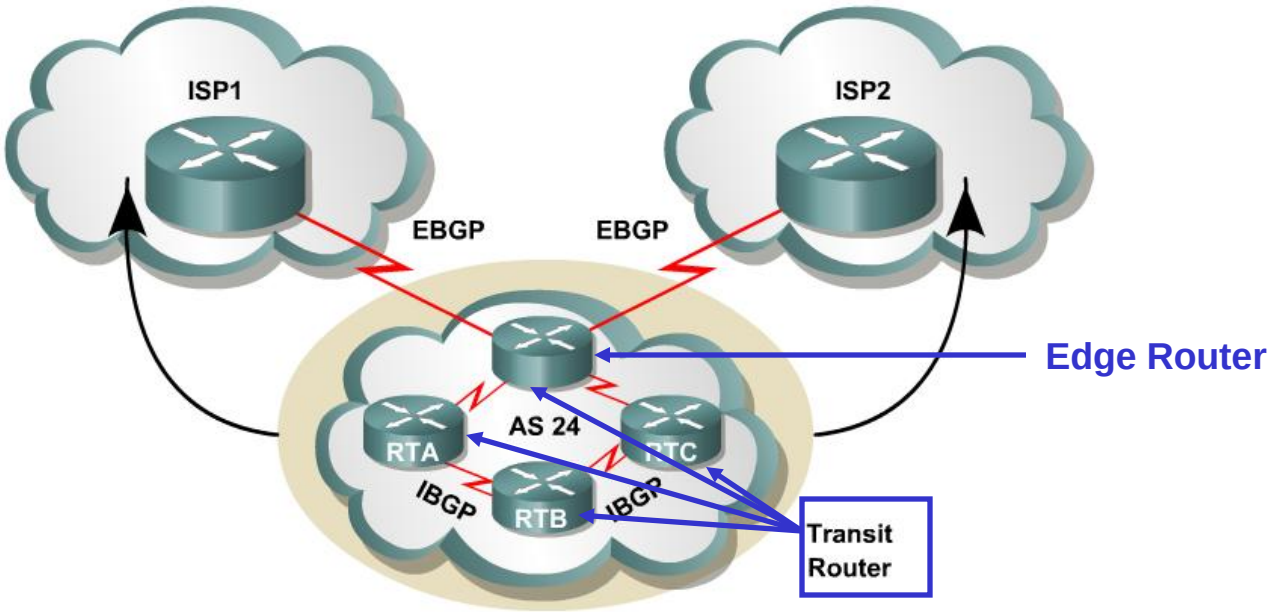
Multi-homed Transit Autonomous Systems



A multihomed transit system can be used for transit traffic by other autonomous systems.

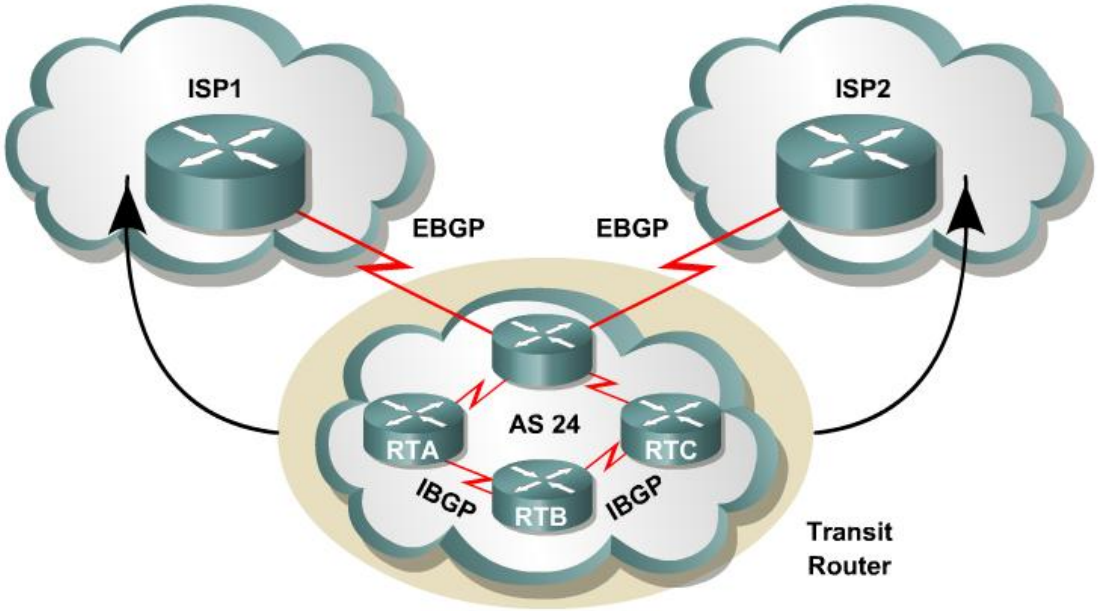
- A multi-homed **transit system** has more than one connection to the outside world and can be used for transit traffic by other autonomous systems.
  - From the point of view of the multi-homed AS, transit traffic is any traffic originating from outside sources bound for outside destinations

Multi-homed Transit Autonomous Systems



- When BGP is running inside an AS, it is referred to as **Internal BGP (IBGP).**
- When BGP runs between autonomous systems, it is called **External BGP (EBGP).**
- If the role of a BGP router is to route IBGP traffic, it is called a **transit router.**
- Routers that sit on the boundary of an AS and that use EBGP to exchange information with the ISP are called **border or edge routers**.

BGP Hazards – Doyle, Routing TCP/IP

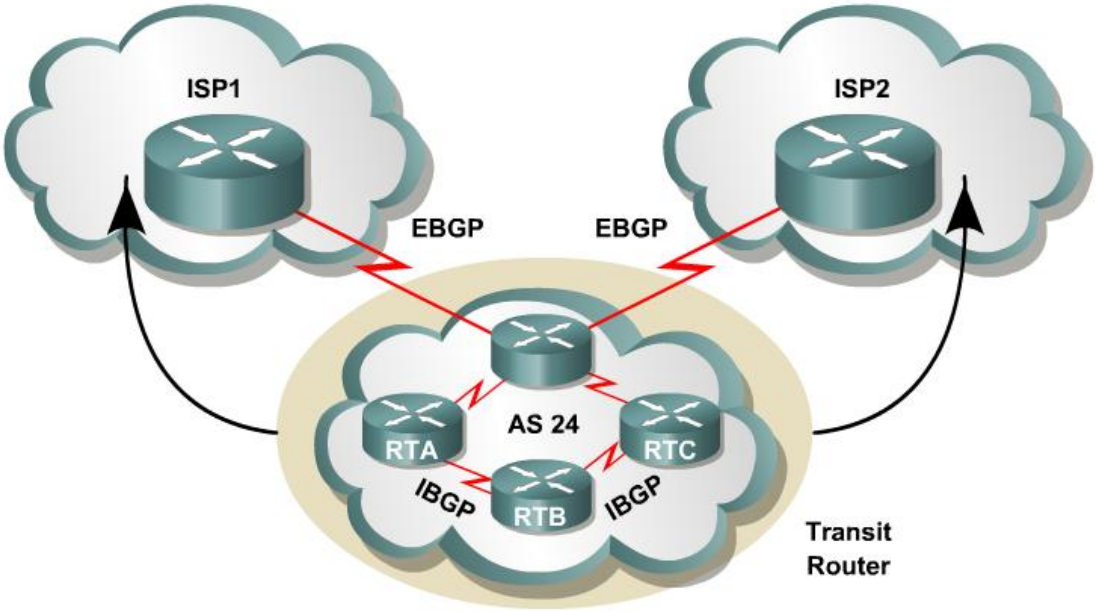


- Creating a BGP “peering” relationship involves an interesting combination of trust and mistrust.
- You must trust the network administrator on that end to know what they are doing.
- At the same time, if you are smart, you will take every practical measure to protect yourself in the event that a mistake is made on the other end.
- “Paranoia is your friend.”

BGP Hazards – Doyle, Routing TCP/IP

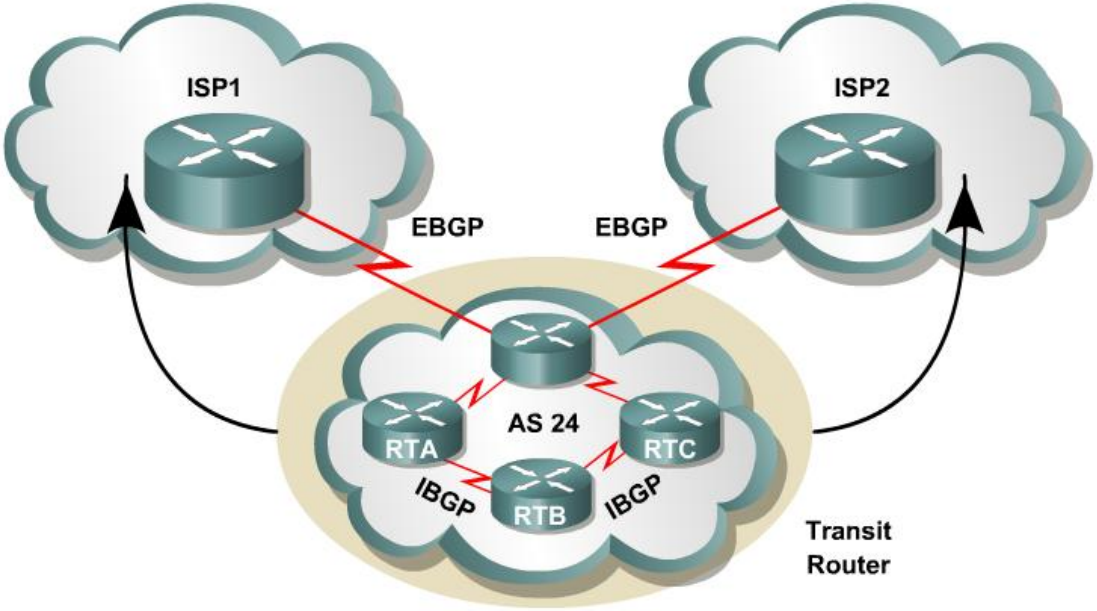
- Your ISP will show little patience with you if you make mistakes in your BGP configuration.
- Suppose, for example, that through some misconfiguration you advertise 207.46.0.0/16 to your ISP.
- On the receiving side, the ISP does not filter out this incorrect route, allowing it to be advertised to the rest of the Internet.
- This particular CIDR block belongs to Microsoft, and you have just claimed to have a route to that destination.
- A significant portion of the Internet community could decide that the best path to Microsoft is through your domain.
- You will receive a flood of unwanted packets across your Internet connection and, more importantly, you will have black-holed traffic that should have gone to Microsoft.
- They will be neither amused nor understanding.

BGP Hazards – Inadvertent Transit Domain



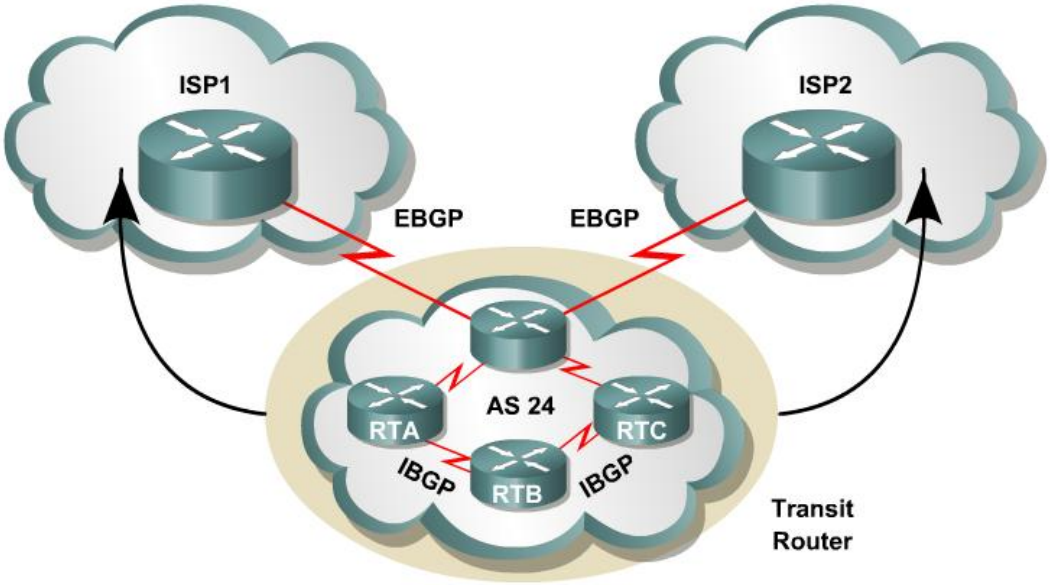
- We inadvertently advertise routes learned from ISP2 to ISP1.
- ISP1 customers will see our network as the best path to ISP2 customers.
- We have become a transit domain for packets from ISP1 to ISP2.

BGP Basics



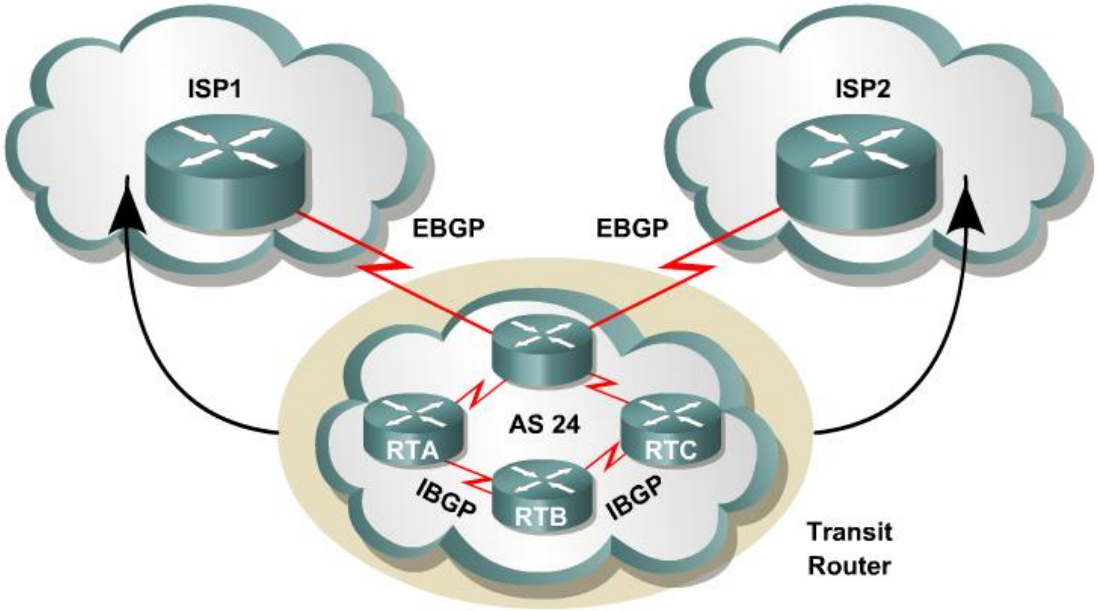
- BGP is a path vector routing protocol.
- Defined in RFC 1772
- BGP is a distance vector routing protocol, in that it relies on downstream neighbors to pass along routes from their routing table.
- BGP uses a list of AS numbers through which a packet must pass to reach a destination.

BGP Basics



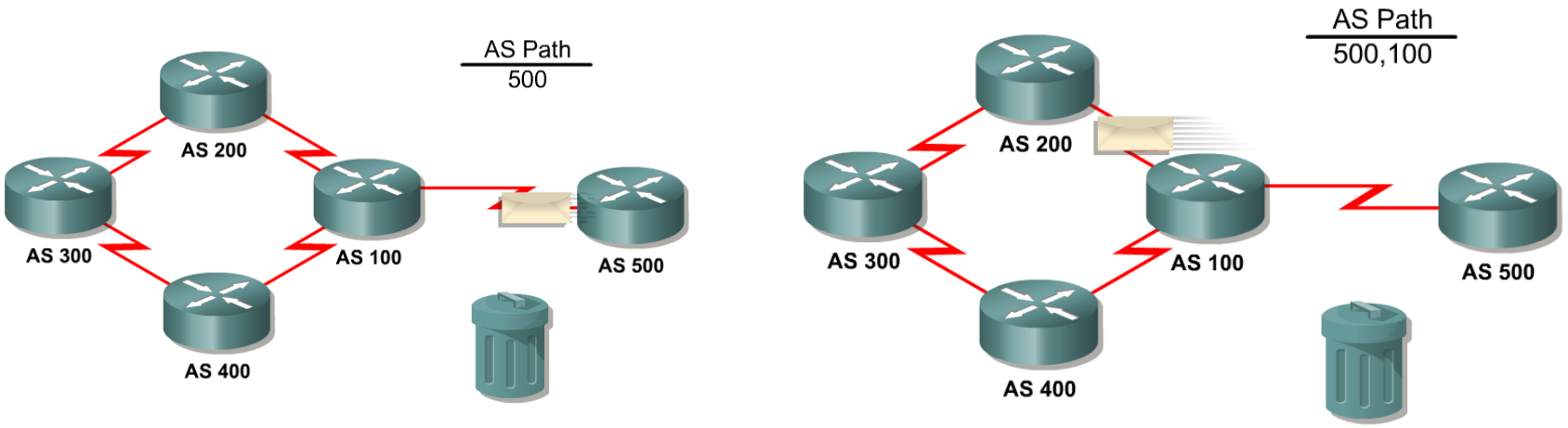
- The function of BGP is to:
  - Exchange routing information between autonomous systems
  - Guarantee the selection of a loop free path.
- BGP4 is the first version of BGP that supports CIDR and route aggregation.
- Common IGPs such as RIP, OSPF, and EIGRP use technical metrics.
  - BGP does not use technical metrics.
- BGP makes routing decisions based on network policies, or rules (later)
- BGP does not show the details of topologies within each AS.
- BGP sees only a tree of autonomous systems.
- Cisco routers maintain a separate routing table to hold BGP routes – **show ip bgp** – later.

BGP Basics



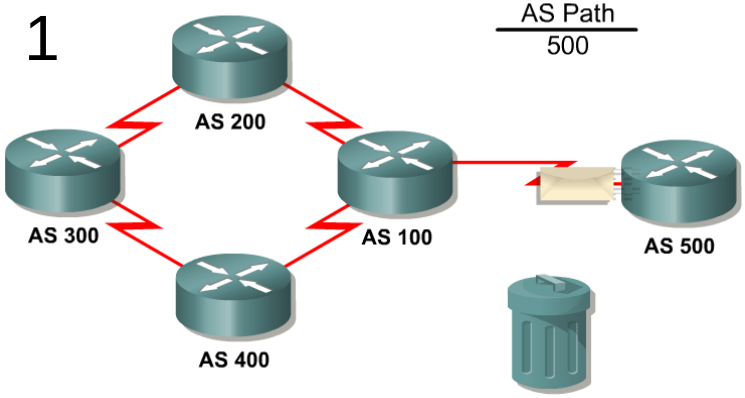
- BGP updates are carried using TCP on port 179.
  - In contrast, RIP updates use UDP port 520
  - OSPF, IGRP, EIGRP does not use a Layer 4 protocol
- Because BGP requires TCP, IP connectivity must exist between BGP peers.
- TCP connections must also be negotiated between them before updates can be exchanged.
- Therefore, BGP inherits those reliable, connection-oriented properties from TCP.

Loop Free Path

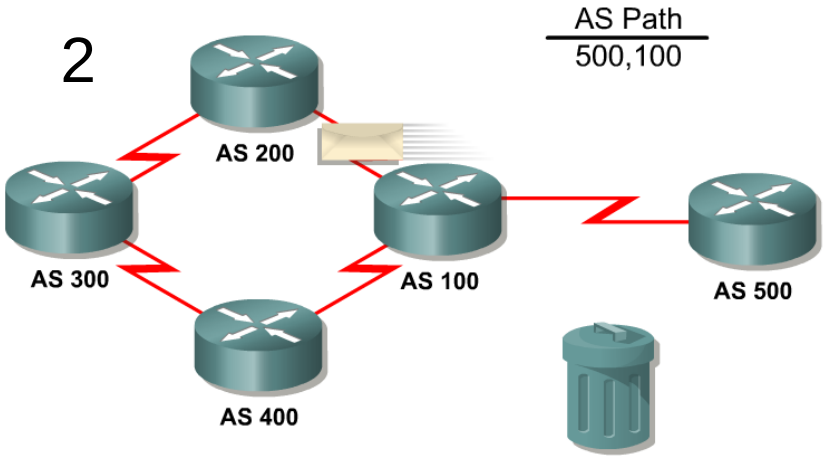


- To guarantee loop free path selection, BGP constructs a graph of autonomous systems based on the information exchanged between BGP neighbors.
- BGP views the whole internetwork as a graph, or tree, of autonomous systems.
- The connection between any two systems forms a path.
- The collection of path information is expressed as a sequence of AS numbers called the AS Path.
- This sequence forms a route to reach a specific destination

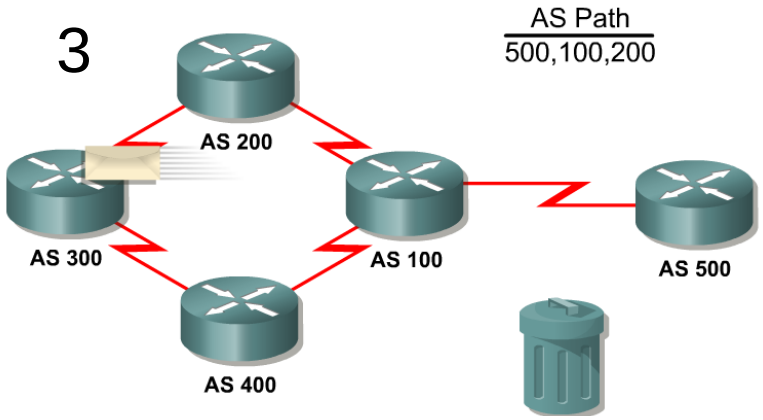
1


AS Path
500
AS 200
AS 300
AS 100
AS 500
AS 400

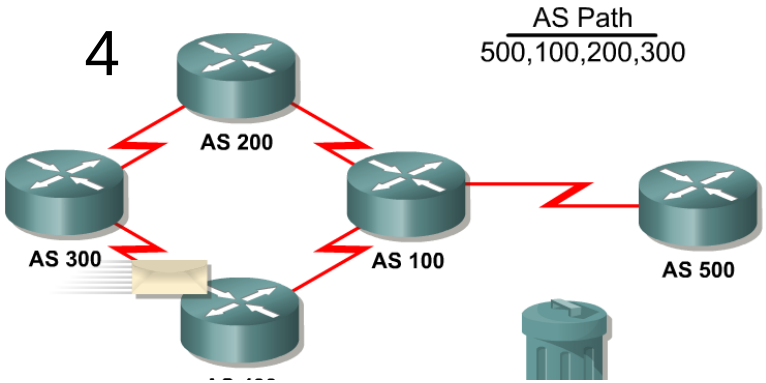
2


AS Path
500,100
AS 200
AS 300
AS 100
AS 500
AS 400

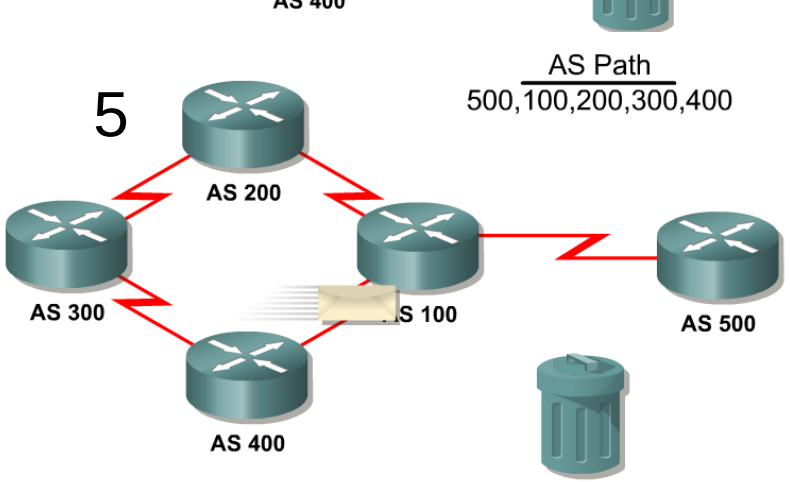
3


AS Path
500,100,200
AS 200
AS 300
AS 100
AS 500
AS 400

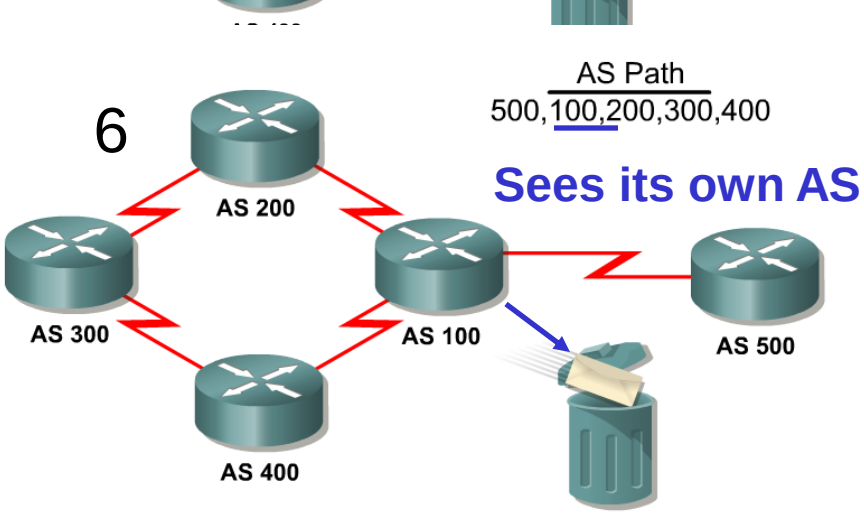
4


AS Path
500,100,200,300
AS 200
AS 300
AS 100
AS 500

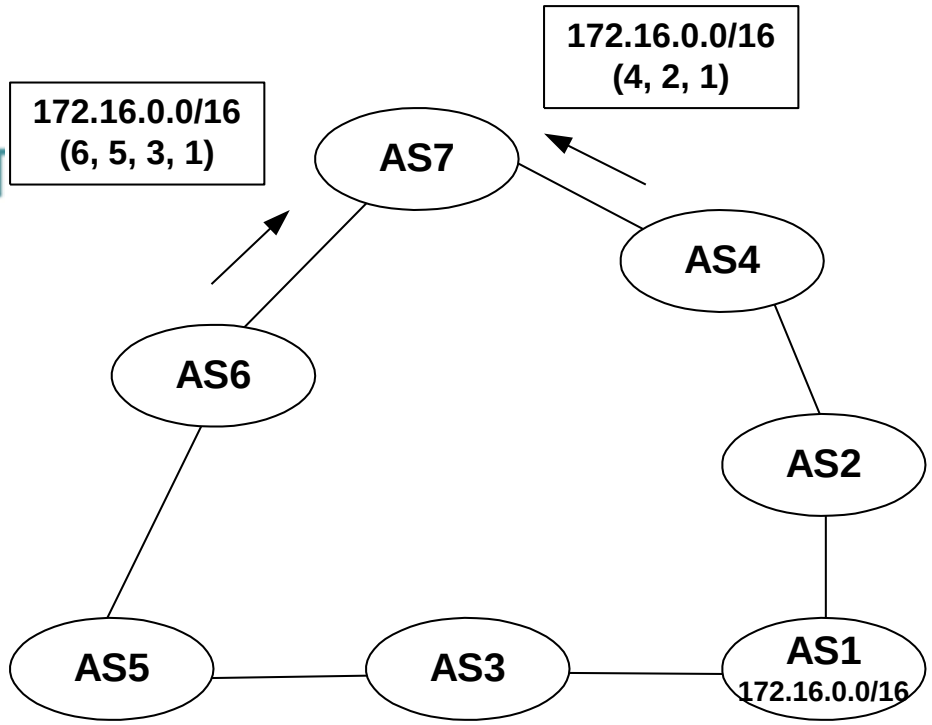
5


AS Path
500,100,200,300,400
AS 200
AS 300
AS 500
AS 400

6

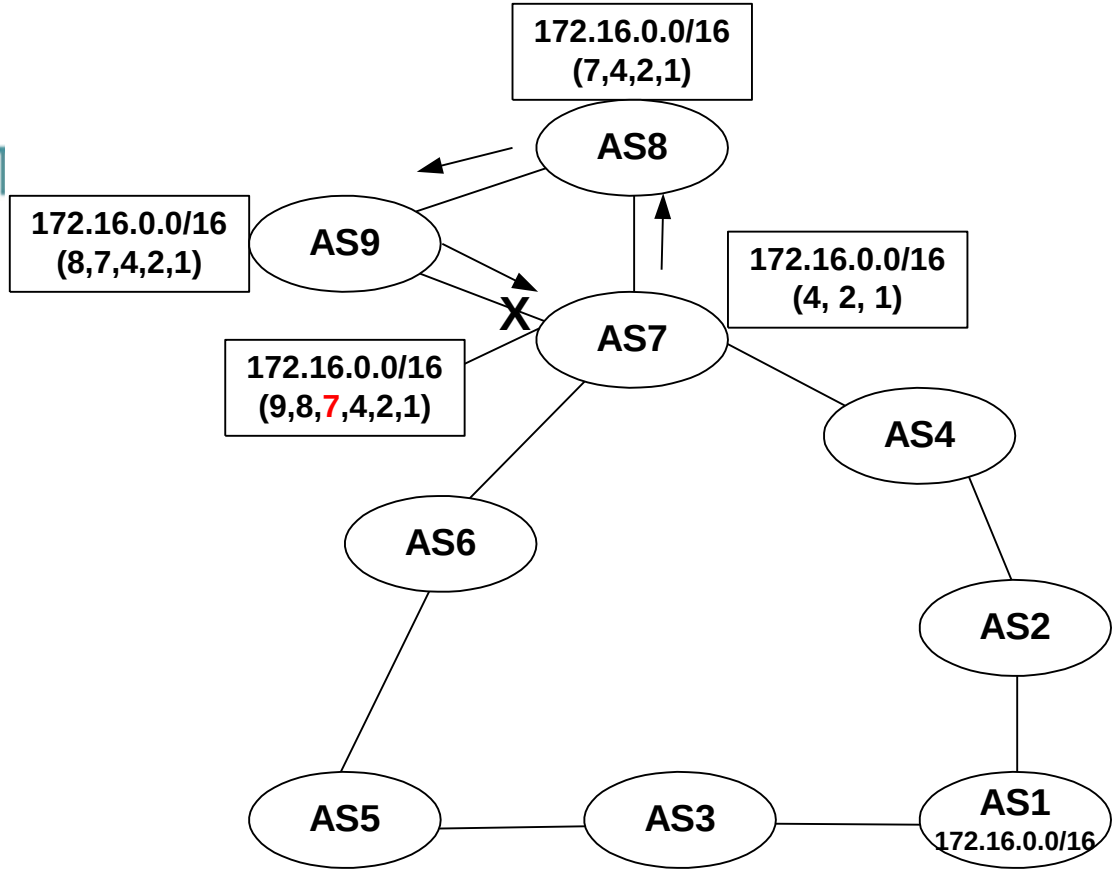

AS Path
500,100,200,300,400
Sees its own AS
AS 200
AS 300
AS 100
AS 500
AS 400

Loop Free Path



- The list of AS numbers associated with a BGP route is called the **AS_PATH** and is one of several path attributes associated with each route.
- Path attributes will be discussed in much more detail later.
- The shortest inter-AS path is very simply determined by the least number of AS numbers.
- All things being equal, BGP prefers routes with shorter AS paths.
- In this example, AS7 will choose the shortest path (4, 2, 1).
- We will see later what happens with equal cost paths.

Loop Free Path



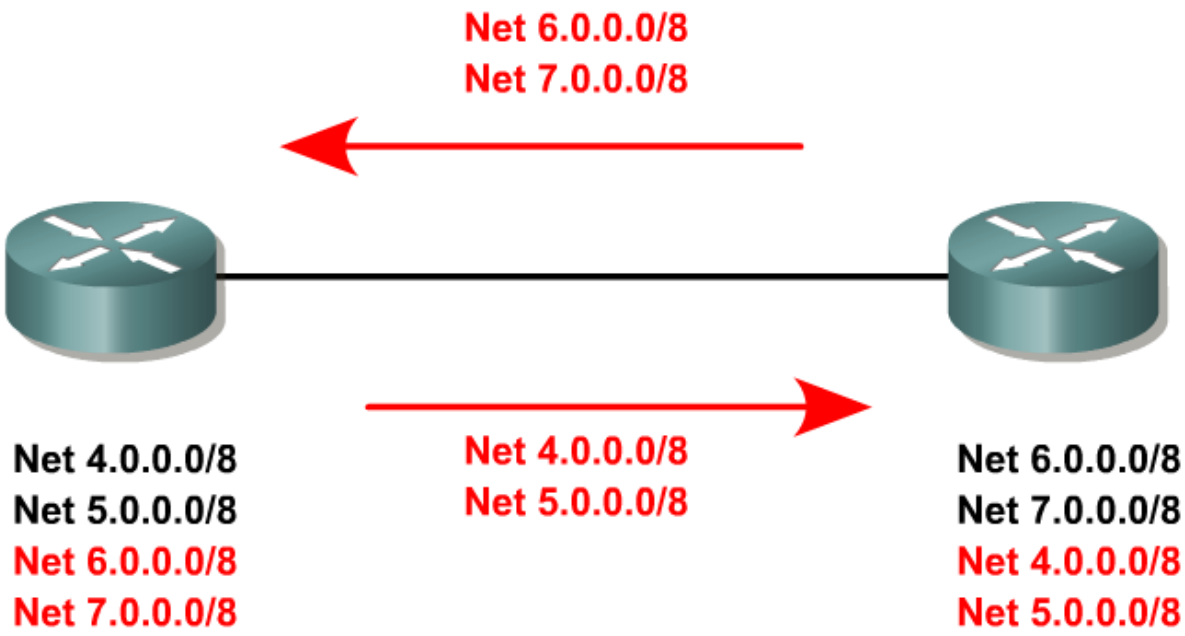
Routing Loop Avoidance

- Route loops can be easily detected when a router receives an update containing its local AS number in the AS_PATH.
- When this occurs, the router will not accept the update, thereby avoiding a potential routing loop.

BGP Operation

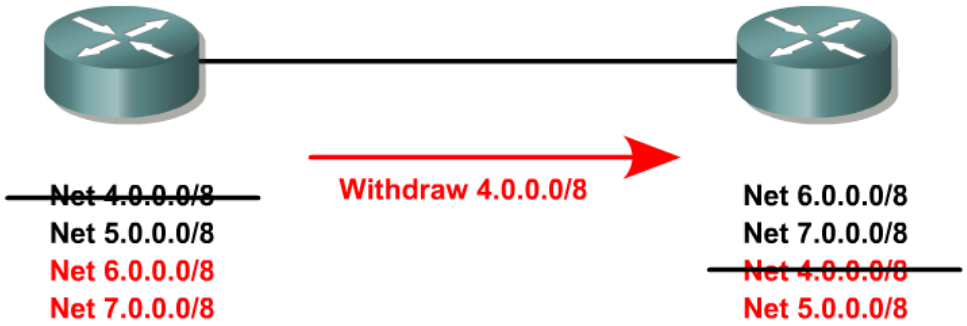
- When two routers establish a TCP-enabled BGP connection between each other, they are called ***neighbors*** or ***peers***.
- Each router running BGP is called a ***BGP speaker***.

Initial Exchange



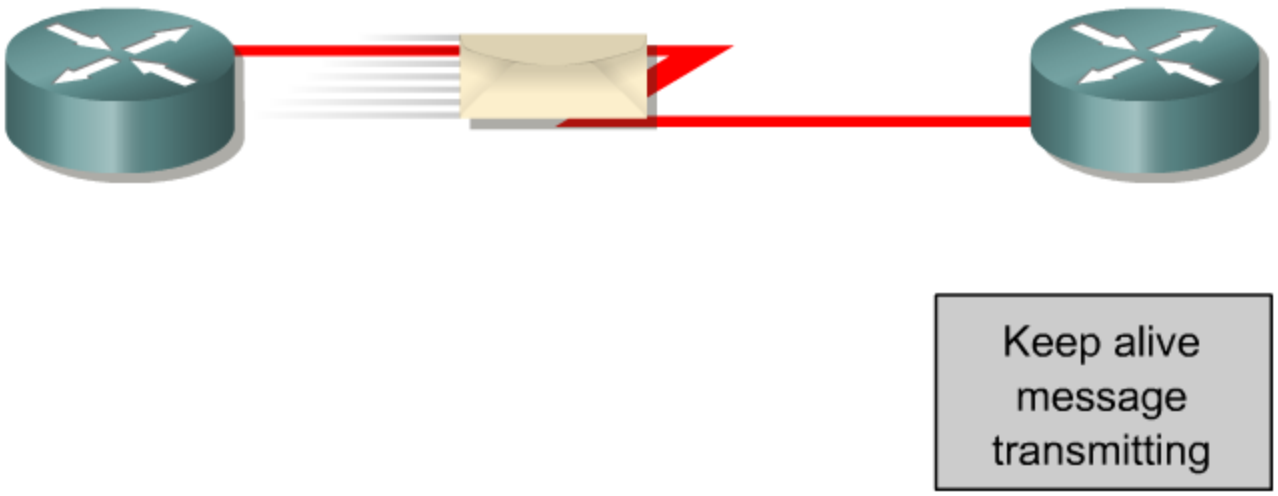
- When BGP neighbors first establish a connection, they exchange all candidate BGP routes.
- After this initial exchange, incremental updates are sent as network information changes.

Withdrawn Routes



- The information for network reachability can change, such as when a route becomes unreachable or a better path becomes available.
- BGP informs its neighbors of this by withdrawing the invalid routes and injecting the new routing information.
- Withdrawn routes are part of the update message. BGP routers keep a table version number that tracks the version of the BGP routing table received from each peer.
- If the table changes, BGP increments the table version number.
- A rapidly incrementing table version is usually an indication of instabilities in the network, or a misconfiguration.

BGP Keepalives



- Peers exchange **keepalive messages** to ensure the connection is maintained.
- The Cisco default keepalive interval is **60 seconds** (RFC 1771 does not specify a standard time).
- If three keepalive intervals (**180 seconds**) pass the peer declares its neighbor down.
- These can be modified with timers **bgp command**.

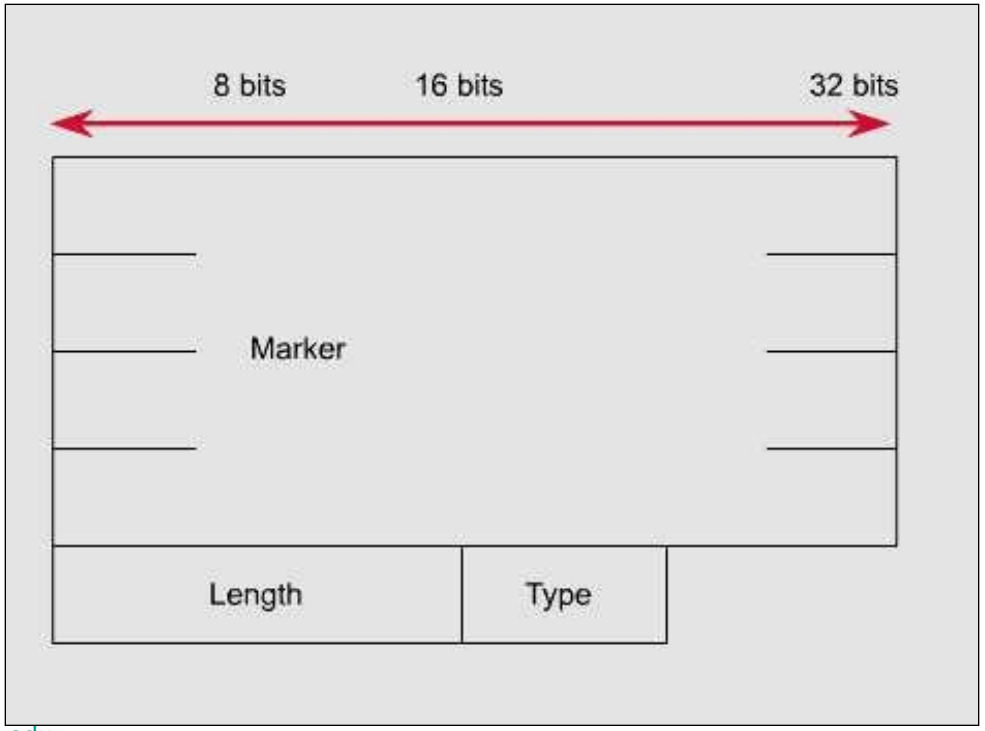
BGP Message Types

- Before establishing a BGP peer connection the two neighbors must perform the standard TCP three-way handshake and open a TCP connection to port 179.
- After the TCP session is established, BGP peers exchanges several messages to open and confirm connection parameters and to send BGP routing information.
- All BGP messages are unicast to the one neighbor over the TCP connection.
- There are four BGP message types:
  - **Type 1: OPEN**
  - **Type 2: KEEPALIVE**
  - **Type 3: UPDATE**
  - **Type 4: NOTIFICATION**

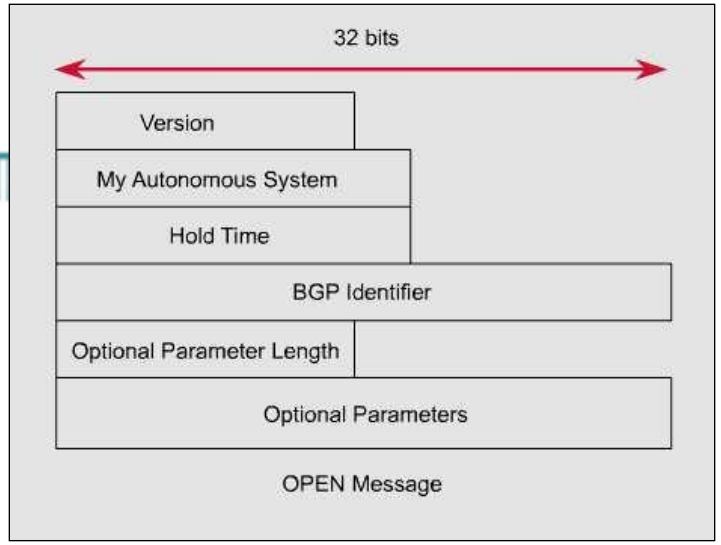
BGP Message Types

Each BGP Message contains the following header:

- **Marker**: The marker field is used to either authenticate incoming BGP messages or to detect loss of synchronization between two BGP peers.
- **Length**: The length field indicates the total BGP message length, including the header.

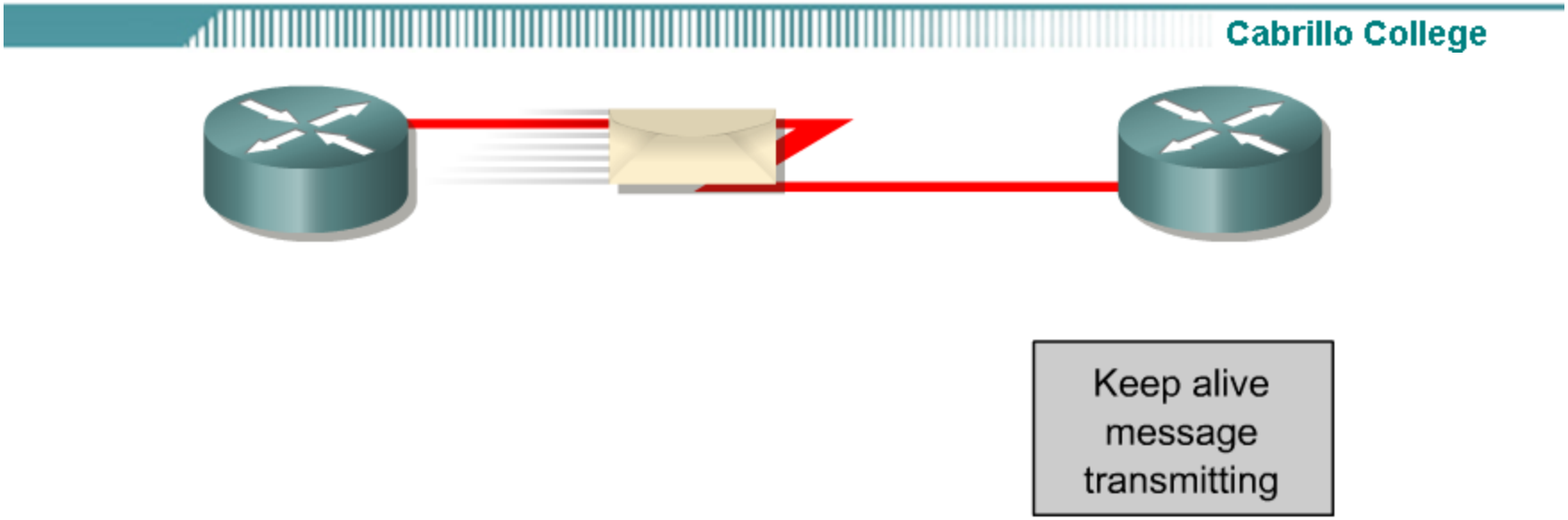


Type 1:
BGP Open Message



- After the TCP session is established, both neighbors send Open messages.
- This message is used to establish connections with peers.
- Each neighbor uses this message to identify itself and to specify its BGP operational parameters including:
  - **BGP version number** (defaults to version 4)
  - **AS number**: AS number of the originating router, determines if BGP session is EBGP or IBGP.
  - **BGP identifier**: IP address that identifies the neighbor using the same method as OSPF router ID.
  - **Optional parameter**: authentication, multiprotocol support and route refresh.

Type 2: BGP Keepalive Message



- This message type is sent periodically between peers to maintain connections and verify paths held by the router sending the keepalive.
- If a router accepts the parameters specified in its neighbor's Open message, it responds with a Keepalive.
- Subsequent Keepalives are **sent every 60 seconds** by Cisco default or equal to one-third the agreed-upon hold time (180 seconds).
- If the periodic timer is set to a value of zero (0), no keepalives are sent.

Type 3: BGP Update Message

- The UPDATE messages contain all the information BGP uses to construct a loop-free picture of the internetwork.
- Update messages advertises feasible routes, withdrawn routes, or both.
- The three basic components of an UPDATE message are:
  - **Network-Layer Reachability Information (NLRI)**
  - **Path Attributes**
  - **Withdrawn Routes**

Type 3: BGP Update Message

Network-Layer Reachability Information (NLRI)

- This is one or more (Length, Prefix) tuples that advertise IP address prefixes and their lengths.
- 192.168.160.0/19
  - Prefix = 192.168.160.0
  - Prefix Length = 19

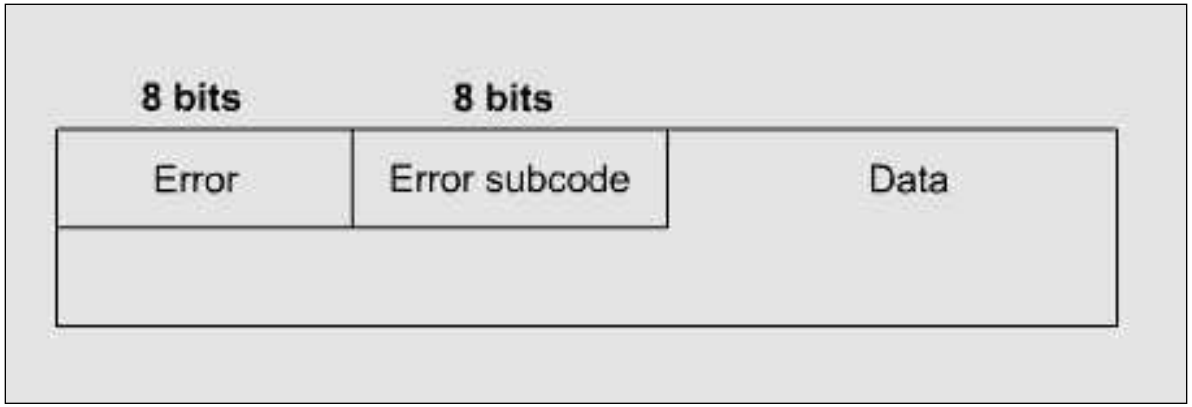
Path Attributes

- This is described later, providing the information that allows BGP to choose a shortest path, detect routing loops, and determine routing policy.

Withdrawn Routes

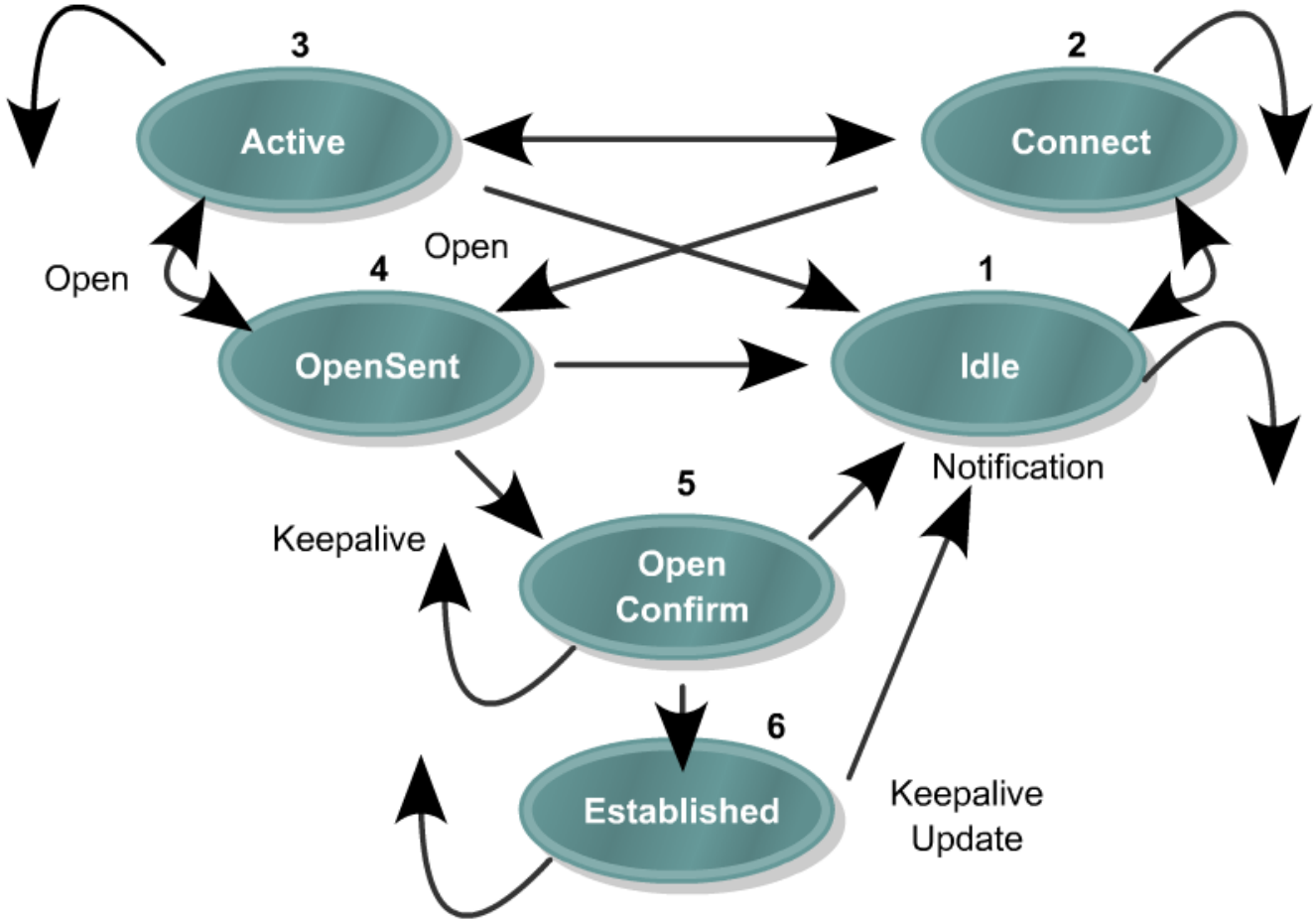
- These are (Length, Prefix) tuples describing destination that have become unreachable and are being withdrawn from service.
- An update message that has no NLRI or path attribute information is used to advertise only routes to be withdrawn from service.

Type 4: BGP Notification Message



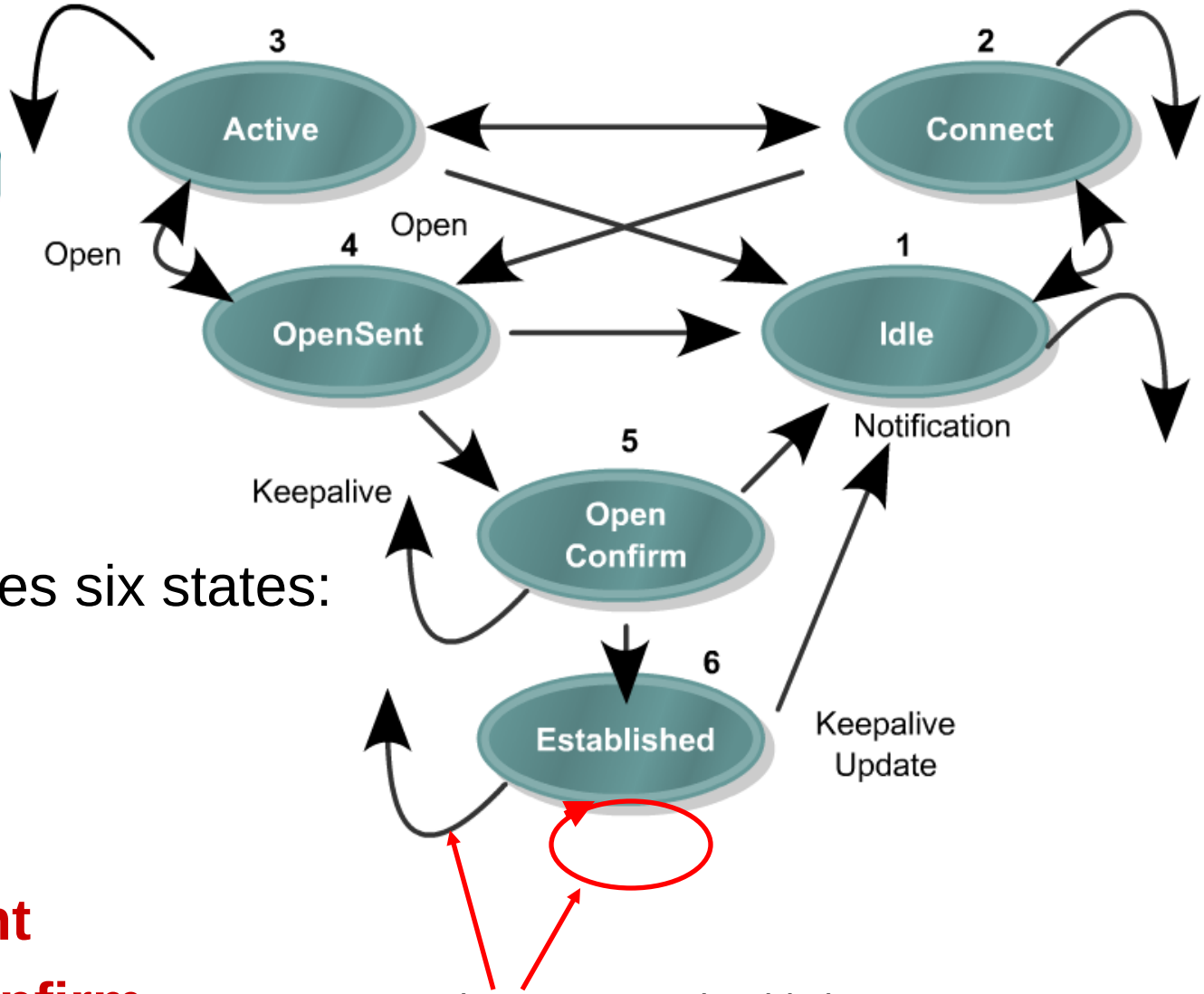
- A NOTIFICATION message is sent whenever an **error** is detected and always causes the BGP connection to close.
- The NOTIFICATION message is composed of the Error Code (8 bits), Error Subcode (8 bits), and a Data fields (variable length).

BGP FSM



- The BGP neighbor negotiation process proceeds through various states, or stages, which can be described in terms of a finite-state machine (FSM).

BGP FSM

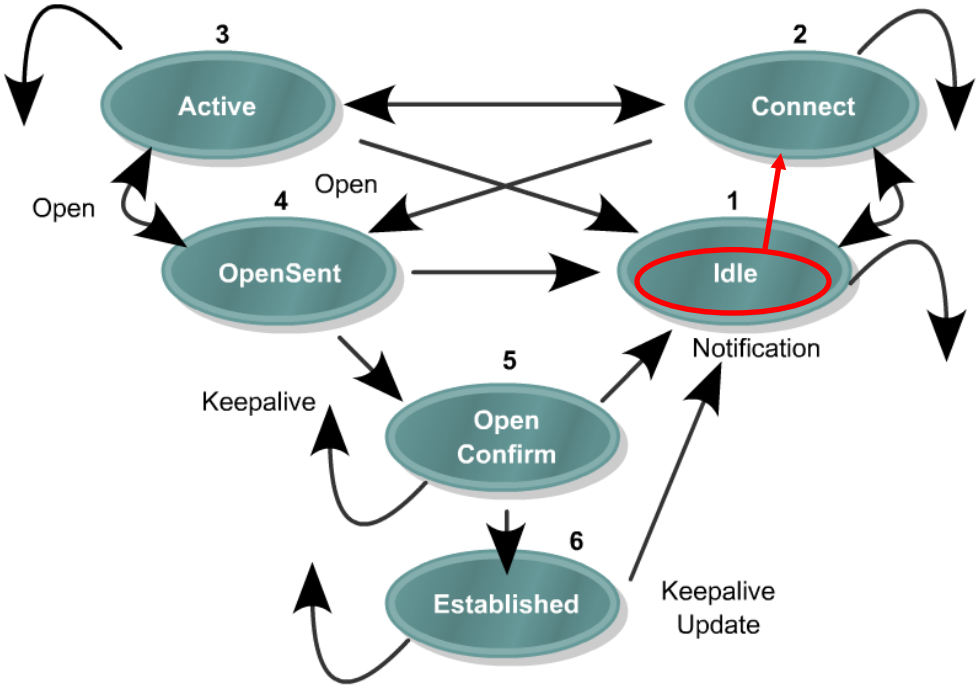


BGP FSM includes six states:

1. **Idle**
2. **Connect**
3. **Active**
4. **OpenSent**
5. **Open Confirm**
6. **Established**

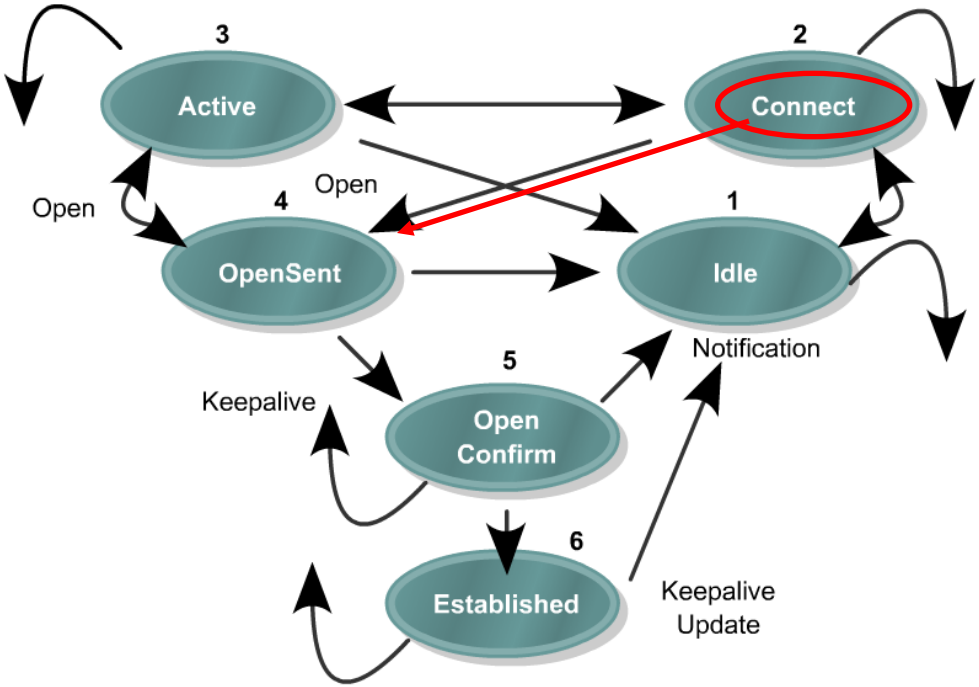
Note: These arrows should show pointing back to the same state.

Idle State



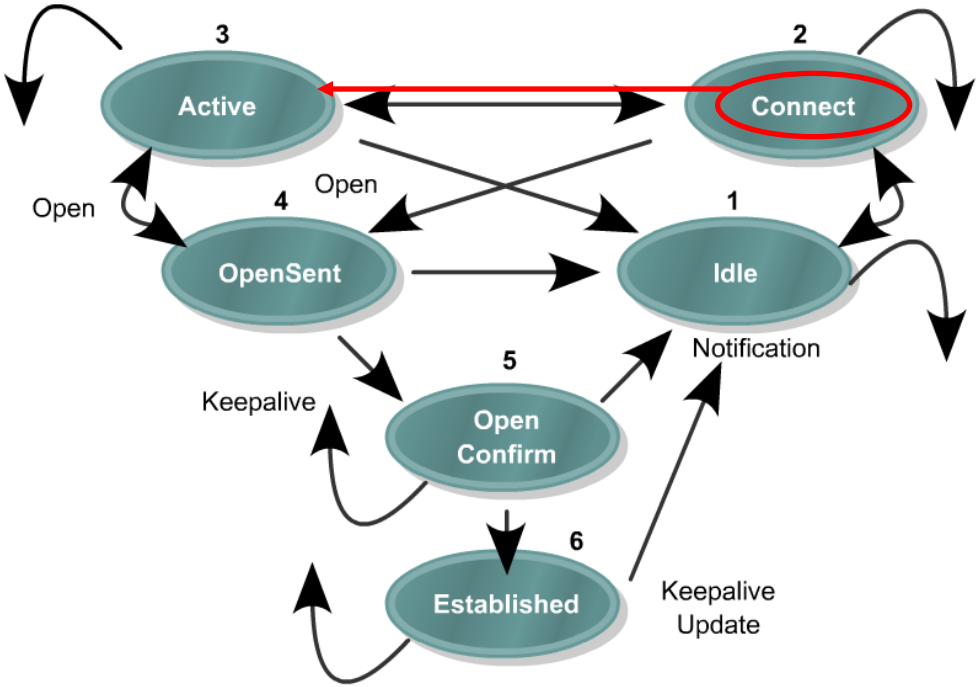
- BGP always begins in the **Idle** state, in which it refuses all incoming connections.
- It is normally initiated by an administrator or a network event.
- When Start event occurs, the BGP process:
  - Initializes all BGP resources
  - Starts the ConnectRetry timer
  - Initializes a TCP connection the the neighbor
  - Listens for a TCP initialization from the neighbor
  - Changes its state to **Connect**

Connect State



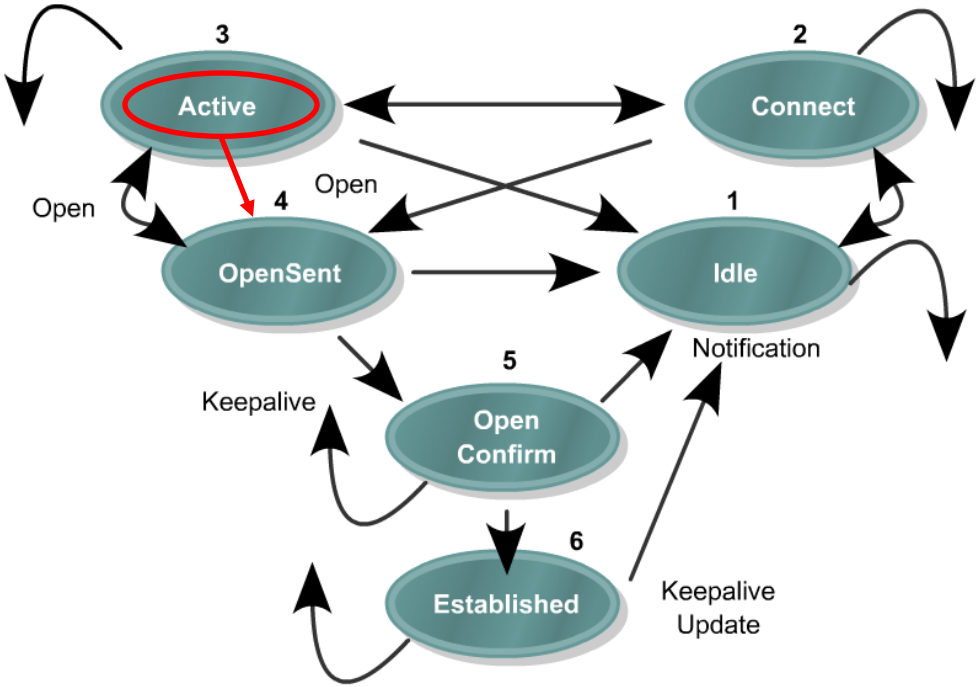
- In this state, the BGP process is waiting for the TCP connection to be completed.
- If the connection is **successful**, the BGP process:
  - Clears the ConnectRetry timer
  - Completes initialization
  - Sends an **Open message** to the neighbor
  - Transitions to the **OpenSent** state

Connect State



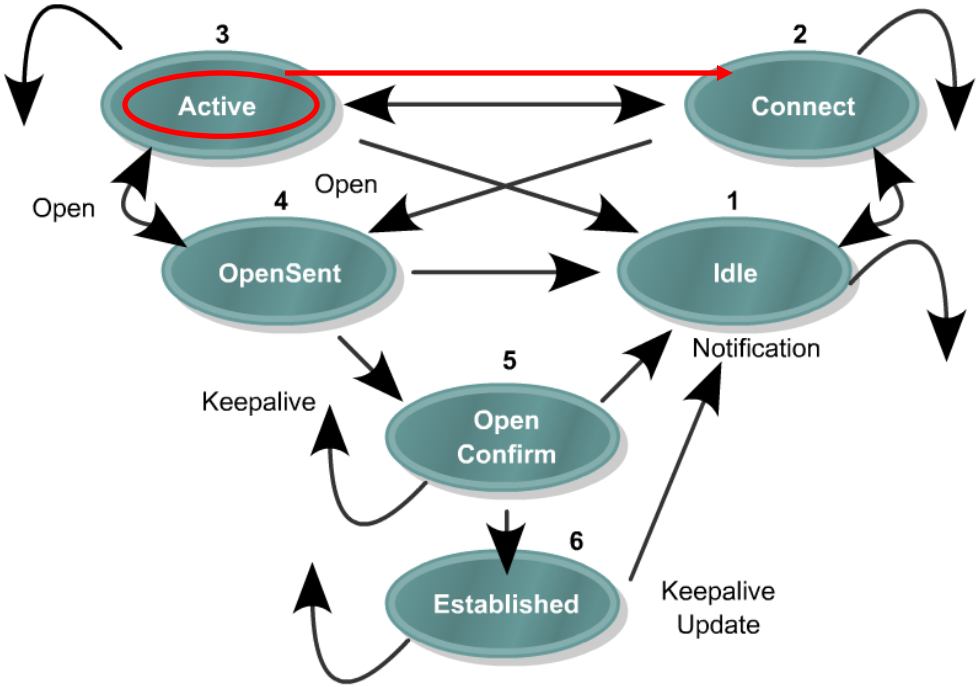
- If the connection is **unsuccessful**, the BGP process:
  - Continues to listen for a connection to be initiated by the neighbor
  - Resets the ConnectRetry timer
  - Transitions to the **Active** state

Active State



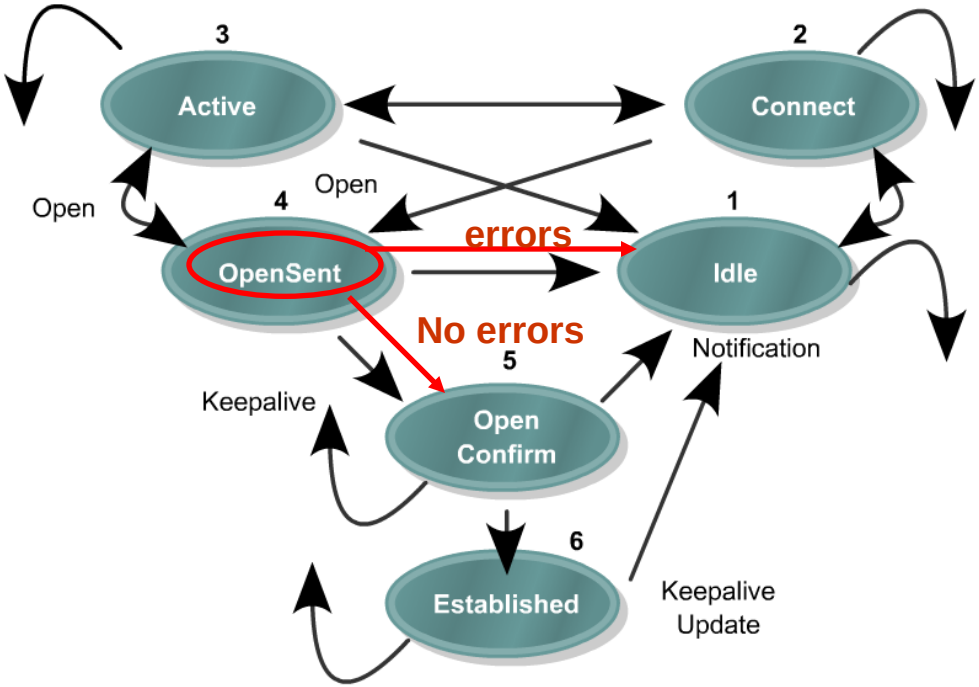
- In this state, the BGP process is trying to initiate a TCP connection with the neighbor.
- If the TCP connection is **successful**:
  - Clears the ConnectRetry timer
  - Completes initialization
  - Sends an **Open message** to the neighbor
  - Transitions to the **OpenSent** state

Active State



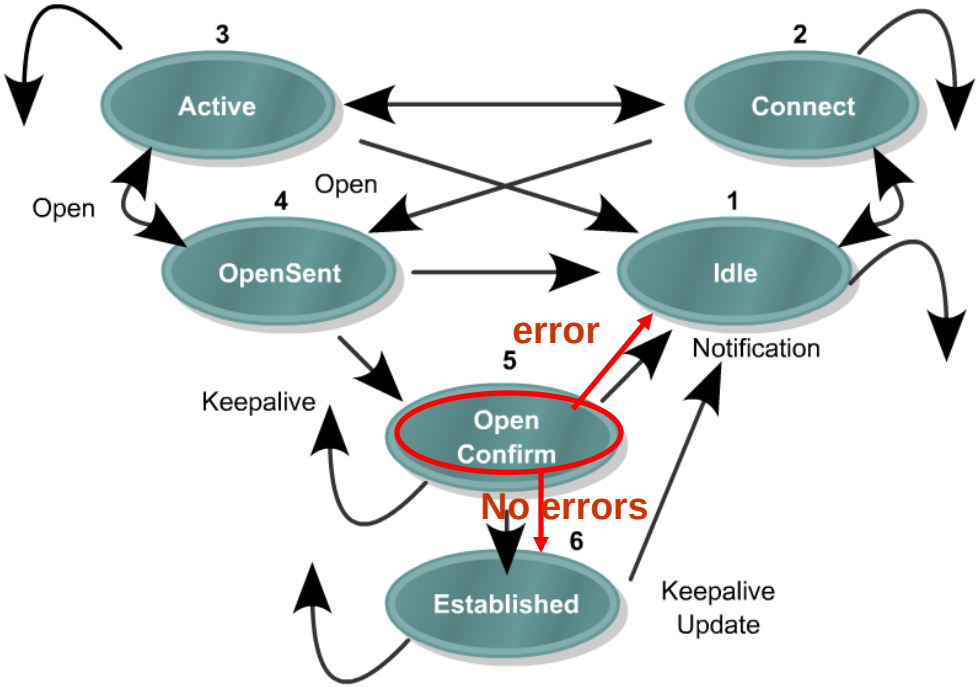
- If the ConnectRetry timer expires while BGP is in the Active State, the BGP process:
  - Transitions back to the **Connect** state
  - Resets the ConnectRetry timer
- In general, a neighbor state that is switching between "**Connect**" and "**Active**" is an indication that something is wrong and that there are problems with the TCP connection.
- It could be because of many TCP retransmissions, or the incapability of a neighbor to reach the IP address of its peer.

OpenSent State



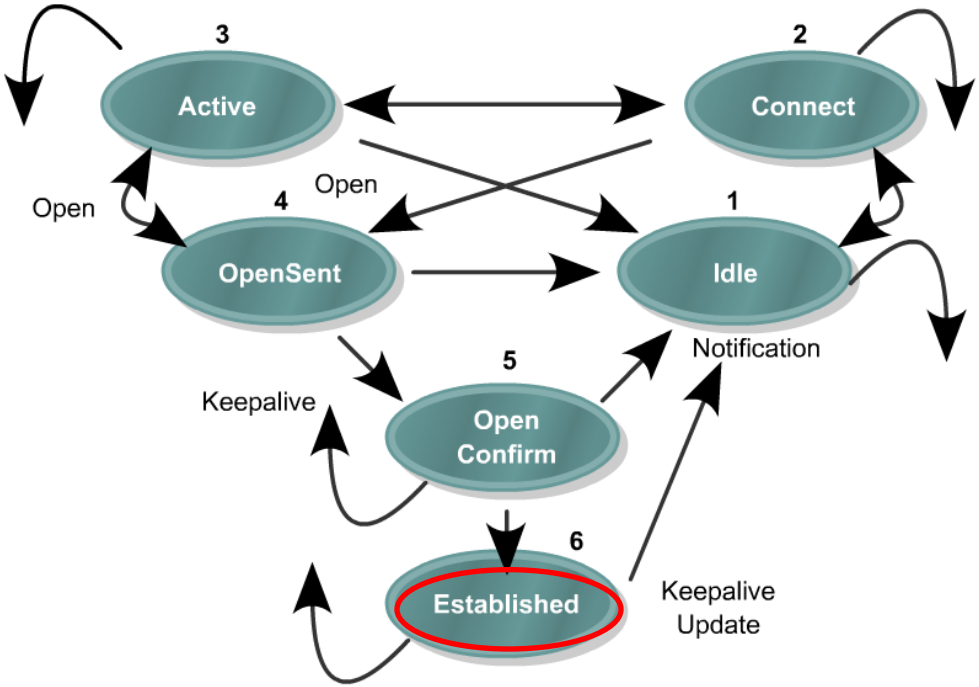
- In this state an **Open message** has been sent and BGP is waiting to hear an Open message from its neighbor.
- When an **Open message** is received, all its fields are checked.
  - **If errors** exist, a **Notification message** is sent and the state transitions to **Idle**.
  - **If no errors** exist, a **Keepalive message** is sent and the Keepalive timer is set, the peer is determined to be internal or external, and state is changed to **OpenConfirm**.

OpenConfirm State



- In this state, the BGP process waits for a **Keepalive** or **Notification message**.
- If a **Keepalive message** is received, the state transitions to **Established**.
- If a **Notification message** is received, or a TCP disconnect is received, the state transitions to **Idle**.

Established State



- In this state, the BGP connection is fully established and the peers can exchange **Update, Keepalive** and **Notification messages**.
- If an **Update** or **Keepalive message** is received, the Hold timer is restarted.
- If a **Notification message** is received, the state transitions to **Idle**.

Path Attributes

Attribute Code	Type
1 — ORIGIN	Well-known mandatory
3 — NEXT_HOP	Well-known mandatory
2 — AS_PATH	Well-known mandatory
4 — MULTI_EXIT_DISC	Optional nontransitive
5 — LOCAL_PREF	Well-known discretionary
6 — ATOMIC_AGGREGATE	Well-known discretionary
7 — AGGREGATOR	Well-known discretionary
8 — COMMUNITY	Optional transitive (Cisco)
9 — ORIGINATOR_ID	Optional nontransitive (Cisco)
10 — Cluster List	Optional nontransitive (Cisco)
11 — Destination Preference	(MCI)
12 — Advertiser	(Baynet)
13 — rcid_path	(Baynet)
255 — Reserved	—

- Much of the work you will do configuring BGP focuses on **path attributes**.
- Each route has its own set of defined attributes, which can include path information, route preference, next-hop, and aggregation information.
- Administrators use these values to enforce **routing policy**.
- Based on attribute values, you can configure BGP to filter routing information, prefer certain paths, or otherwise customize its behavior.
- Every **UPDATE message** has a variable-length sequence of path attributes in the form <attribute type, attribute length, attribute value>.

Path Attributes

Attribute Code	Type
1 — ORIGIN	Well-known mandatory
3 — NEXT_HOP	Well-known mandatory
2 — AS_PATH	Well-known mandatory
4 — MULTI_EXIT_DISC	Optional nontransitive
5 — LOCAL_PREF	Well-known discretionary
6 — ATOMIC_AGGREGATE	Well-known discretionary
7 — AGGREGATOR	Well-known discretionary
8 — COMMUNITY	Optional transitive (Cisco)
9 — ORIGINATOR_ID	Optional nontransitive (Cisco)
10 — Cluster List	Optional nontransitive (Cisco)
11 — Destination Preference	(MCI)
12 — Advertiser	(Baynet)
13 — rcid_path	(Baynet)
255 — Reserved	—

- Since you will use path attributes extensively when configuring routing policy, you should note that not all vendor implementations of BGP recognize the same attributes.
- In fact, path attributes come in four different types:
  - **Well-known mandatory**
  - **Well-known discretionary**
  - **Optional transitive**
  - **Optional non-transitive**

Path Attributes

Attribute Code	Type
1 — ORIGIN	Well-known mandatory
3 — NEXT_HOP	Well-known mandatory
2 — AS_PATH	Well-known mandatory
4 — MULTI_EXIT_DISC	Optional nontransitive
5 — LOCAL_PREF	Well-known discretionary
6 — ATOMIC_AGGREGATE	Well-known discretionary
7 — AGGREGATOR	Well-known discretionary
8 — COMMUNITY	Optional transitive (Cisco)
9 — ORIGINATOR_ID	Optional nontransitive (Cisco)
10 — Cluster List	Optional nontransitive (Cisco)
11 — Destination Preference	(MCI)
12 — Advertiser	(Baynet)
13 — rcid_path	(Baynet)
255 — Reserved	—

Well-known mandatory

- **An attribute that has to exist in the BGP UPDATE packet**.
- It must be recognized by all BGP implementations.
- If a well-known attribute is missing, a notification error will be generated; this ensures that all BGP implementations agree on a standard set of attributes.

Example: AS_PATH attribute.

Path Attributes

Attribute Code	Type
1 — ORIGIN	Well-known mandatory
3 — NEXT_HOP	Well-known mandatory
2 — AS_PATH	Well-known mandatory
4 — MULTI_EXIT_DISC	Optional nontransitive
5 — LOCAL_PREF	Well-known discretionary
6 — ATOMIC_AGGREGATE	Well-known discretionary
7 — AGGREGATOR	Well-known discretionary
8 — COMMUNITY	Optional transitive (Cisco)
9 — ORIGINATOR_ID	Optional nontransitive (Cisco)
10 — Cluster List	Optional nontransitive (Cisco)
11 — Destination Preference	(MCI)
12 — Advertiser	(Baynet)
13 — rcid_path	(Baynet)
255 — Reserved	—

Well-known discretionary

- An attribute that is **recognized by all BGP implementations**
- But may or **may not be sent** in the BGP UPDATE message.

Example: LOCAL_PREF

Path Attributes

Attribute Code	Type
1 — ORIGIN	Well-known mandatory
3 — NEXT_HOP	Well-known mandatory
2 — AS_PATH	Well-known mandatory
4 — MULTI_EXIT_DISC	Optional nontransitive
5 — LOCAL_PREF	Well-known discretionary
6 — ATOMIC_AGGREGATE	Well-known discretionary
7 — AGGREGATOR	Well-known discretionary
8 — COMMUNITY	Optional transitive (Cisco)
9 — ORIGINATOR_ID	Optional nontransitive (Cisco)
10 — Cluster List	Optional nontransitive (Cisco)
11 — Destination Preference	(MCI)
12 — Advertiser	(Baynet)
13 — rcid_path	(Baynet)
255 — Reserved	—

Optional transitive

- An attribute that may or may not be, recognized by all BGP implementations (thus, **optional**).
- Because the attribute is **transitive**, BGP **should accept and advertise** the attribute even if it isn't recognized.

Example: COMMUNITY

Path Attributes

Attribute Code	Type
1 — ORIGIN	Well-known mandatory
3 — NEXT_HOP	Well-known mandatory
2 — AS_PATH	Well-known mandatory
4 — MULTI_EXIT_DISC	Optional nontransitive
5 — LOCAL_PREF	Well-known discretionary
6 — ATOMIC_AGGREGATE	Well-known discretionary
7 — AGGREGATOR	Well-known discretionary
8 — COMMUNITY	Optional transitive (Cisco)
9 — ORIGINATOR_ID	Optional nontransitive (Cisco)
10 — Cluster List	Optional nontransitive (Cisco)
11 — Destination Preference	(MCI)
12 — Advertiser	(Baynet)
13 — rcid_path	(Baynet)
255 — Reserved	—

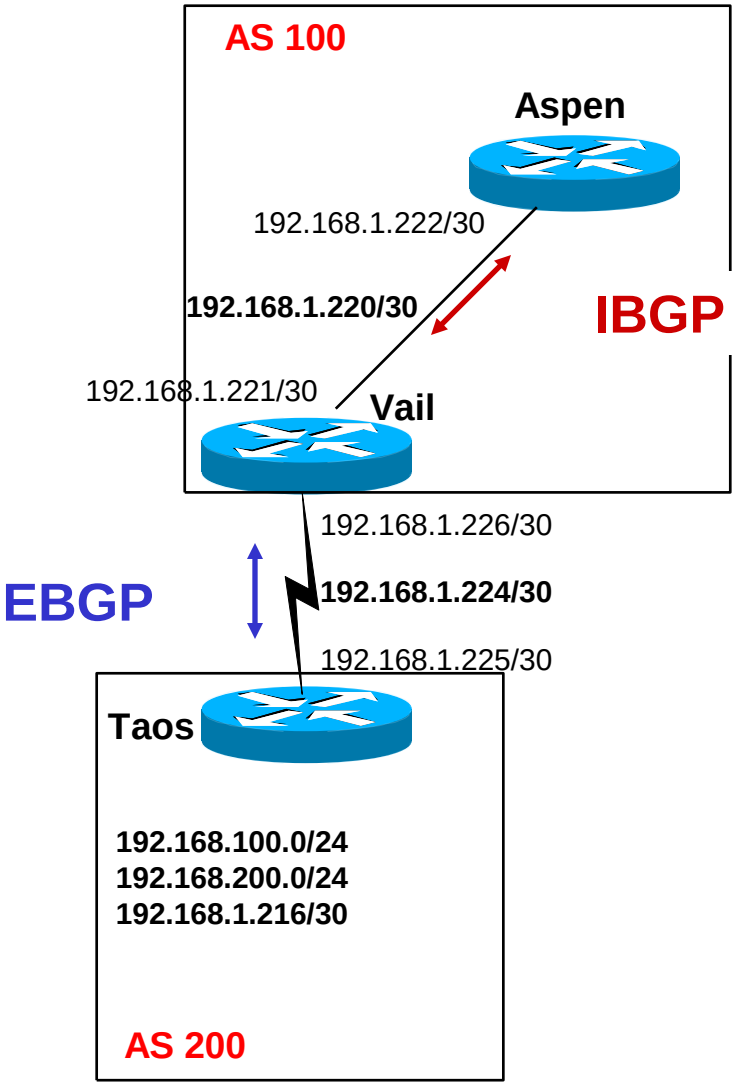
Optional non-transitive

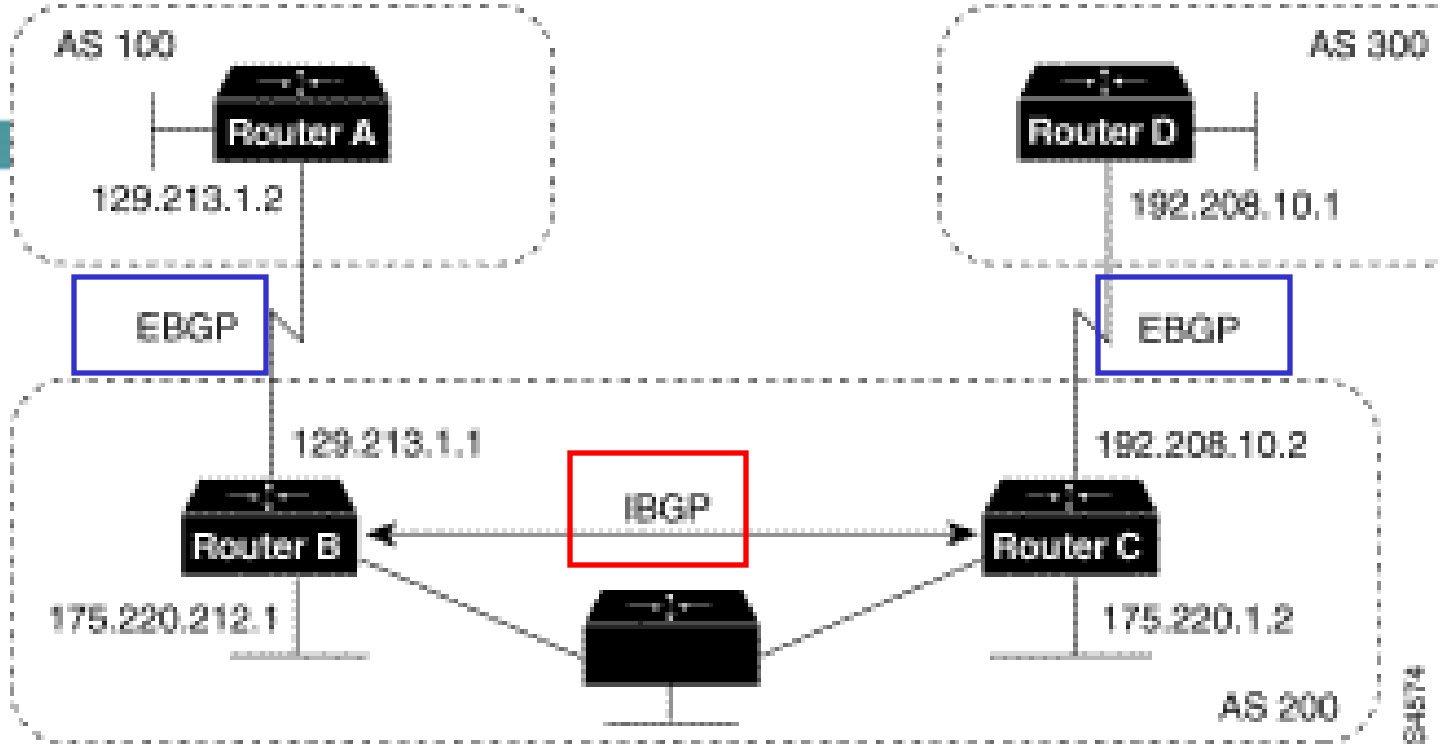
- An attribute that **may or may not be, recognized** by all BGP implementations.
- Whether or not the receiving BGP router recognizes the attribute, it is **non-transitive**, and **should not be passed along to other BGP peers**.

Example: ORIGINATOR_ID

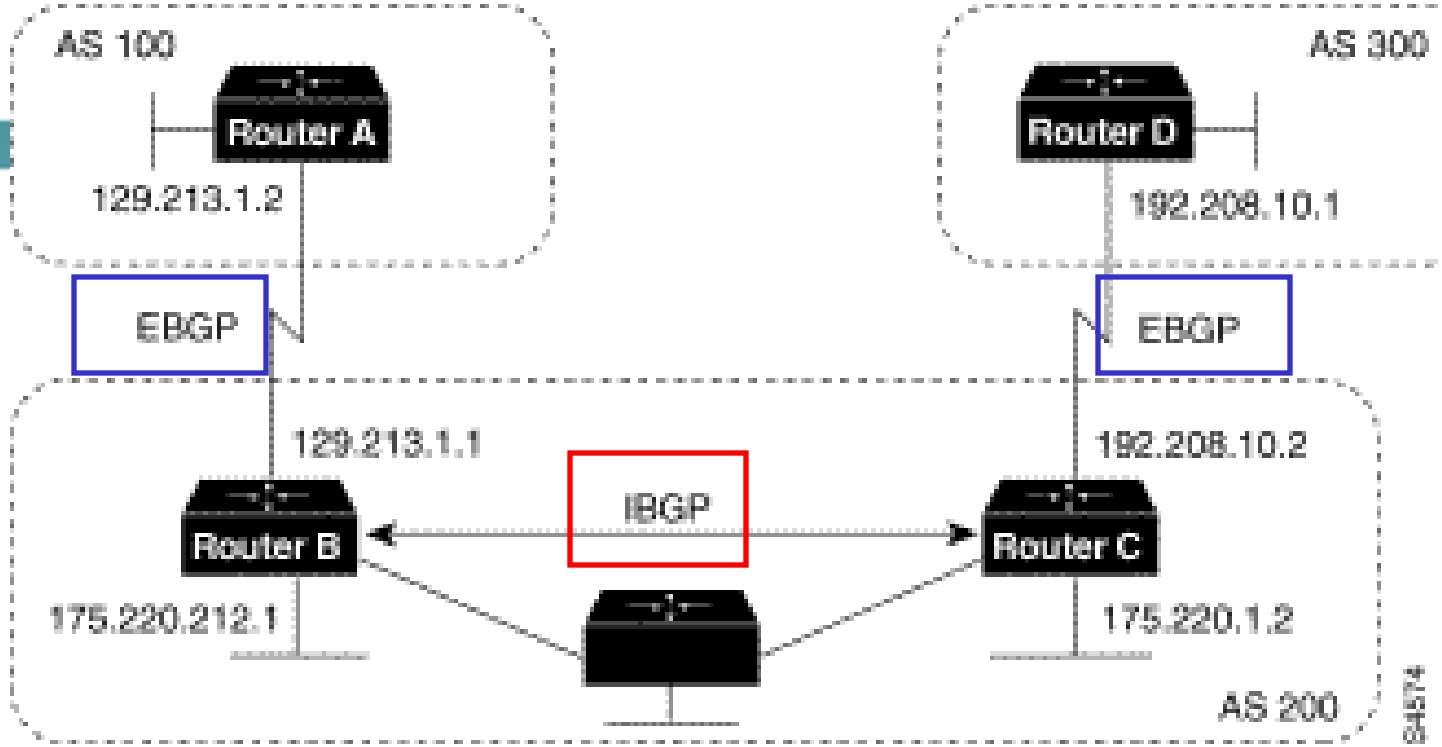
IBGP vs EBGP

- When BGP is running inside an AS, it is referred to as **Internal BGP (IBGP)**.
  - If a BGP router's role is to route IBGP traffic, it is called a transit router.
- When BGP runs between autonomous systems, it is called **External BGP (EBGP)**.
  - Routers that sit on the boundary of an AS and use EBGP to exchange information with the ISP are called border routers.
- "With very few exceptions, interior BGP (IBGP) – BGP between peers in the same AS – is used only in multihomed scenarios." – Doyle



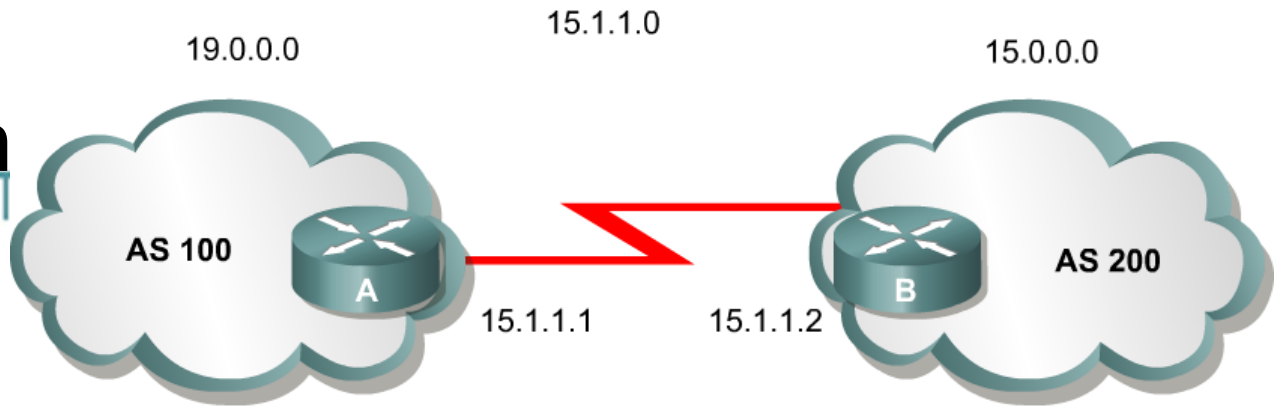


- Routers A and B are running **EBGP (BGP)**, and Routers B and C are running **IBGP**.
- Note that the **EBGP (BGP)** peers are directly connected and that the **IBGP** peers are not. (They can be.)
- As long as there is an **IGP** running that allows the two neighbors to reach one another, IBGP peers do not have to be directly connected.
- ***More later!***



- All **BGP** speakers within an AS must establish a peer relationship with each other, that is, the **BGP** speakers within an AS must be fully meshed logically. (later)
- BGP4 provides two techniques that alleviate the requirement for a logical full mesh: confederations and route reflectors. (later)
- AS 200 is a ***transit AS*** for AS 100 and AS 300---that is, AS 200 is used to transfer packets between AS 100 and AS 300.

BGP Configuration



Configuration for A

```
router bgp 100
network 19.0.0.0
neighbor 15.1.1.2 remote-as 200
```

Configuration for B

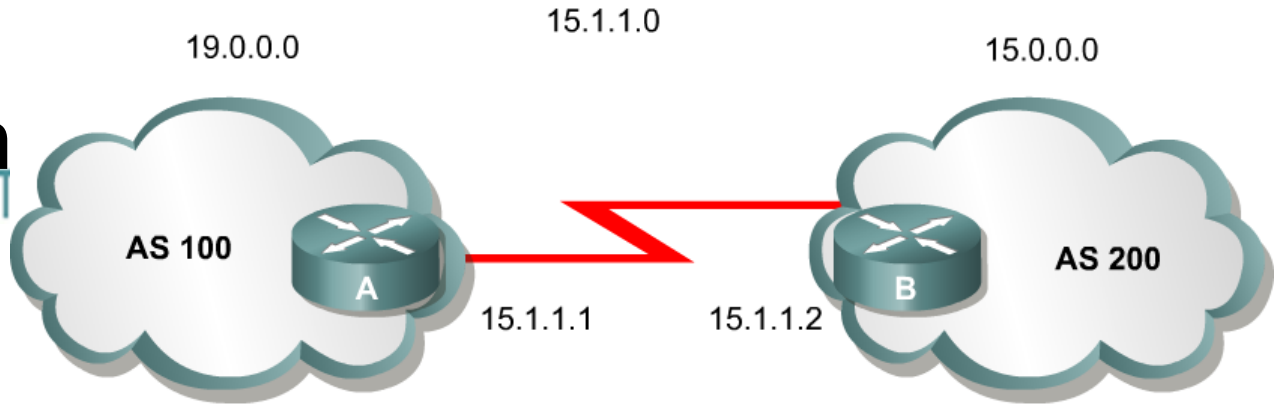
```
router bgp 200
network 15.0.0.0
neighbor 15.1.1.1 remote-as 100
```

- To begin configuring a BGP process, issue the following familiar command:

  `Router(config)#`**`router bgp`** ***`AS-number`***

- BGP configuration commands appear on the surface to mirror the syntax of familiar IGP (for example, RIP, OSPF) commands.
- Although the syntax is similar, the function of these commands is significantly different.
- **Note**: Cisco IOS permits only one BGP process to run at a time, thus, **a router cannot belong to more than one AS.**

BGP Configuration



Configuration for A

```
router bgp 100
network 19.0.0.0
neighbor 15.1.1.2 remote-as 200
```

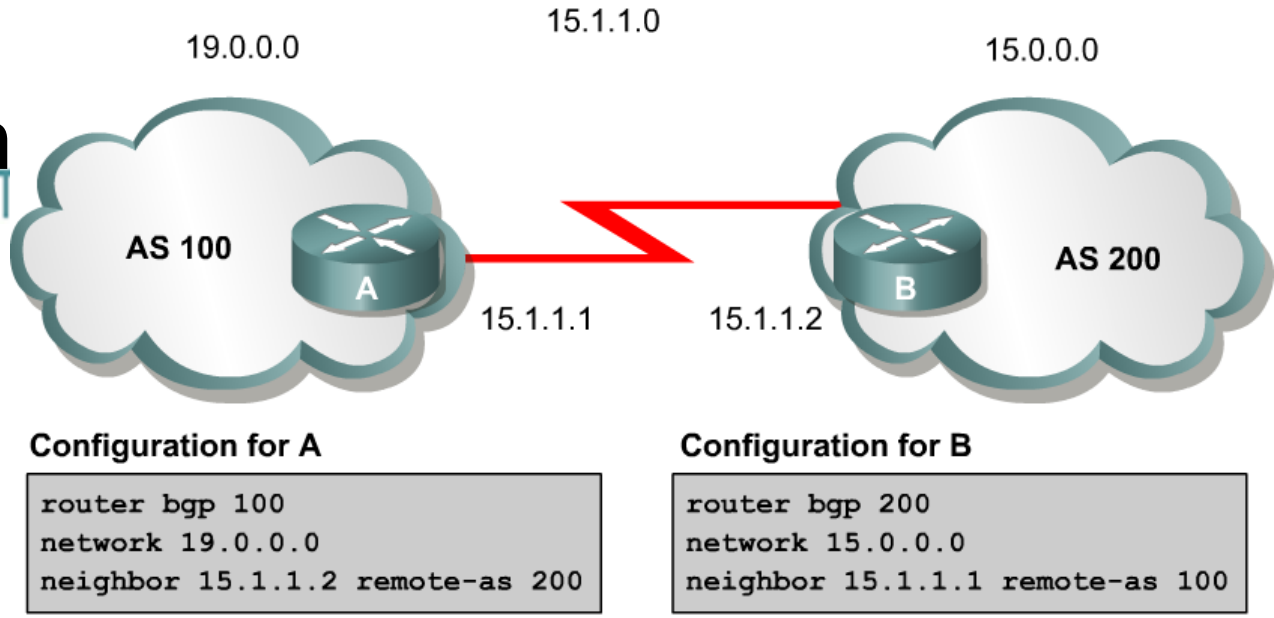
Configuration for B

```
router bgp 200
network 15.0.0.0
neighbor 15.1.1.1 remote-as 100
```

`Router(config-router)#`**`network network-number [mask network-mask]`**

- The **network** command is used with **IGPs**, such as RIP, to determine the interfaces on which to send and receive updates, as well as which directly connected networks to advertise.
- However, when configuring **BGP**, the **network** command does **not** affect what interfaces BGP runs on.
- In BGP, the **network** command tells the BGP process **what locally learned networks to advertise**.
- The networks can be **connected routes, static routes, or routes learned via a dynamic routing protocol**, such as RIP.
  - Thus, configuring just a **network** statement will **not** establish a BGP neighbor relationship. This is a major difference between BGP and IGPs.

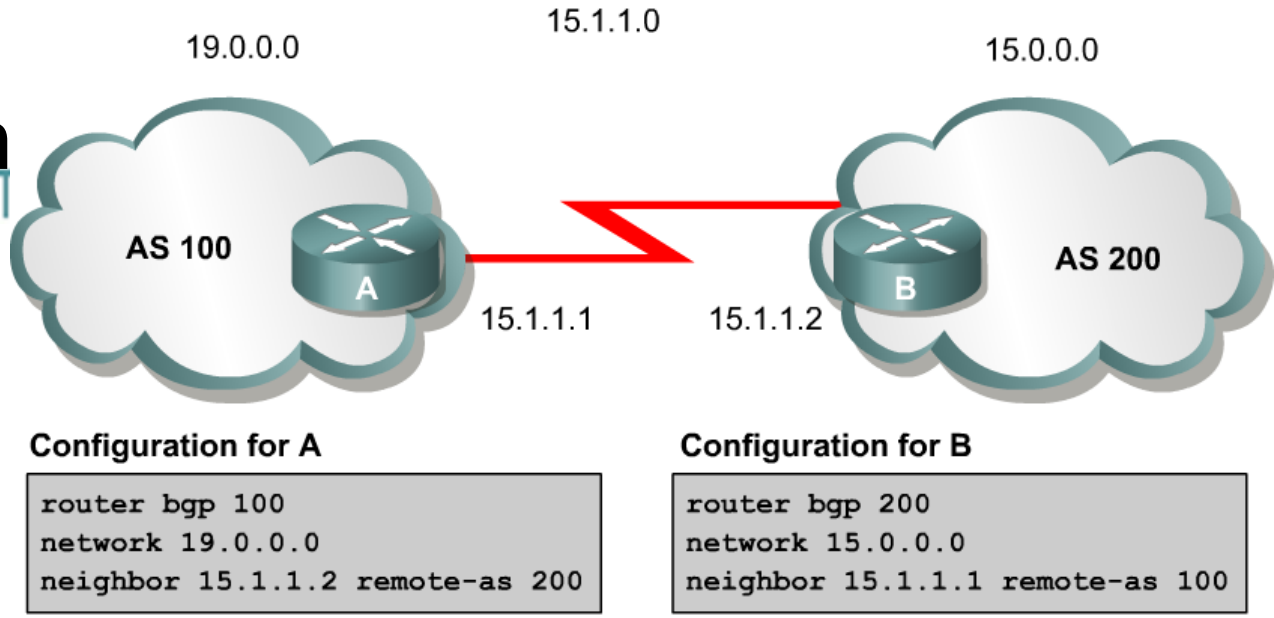
BGP Configuration



network command continued…

- ***These networks must also exist in the local router's routing table (show ip route), or they will not be sent out in updates.***
- You can use the **mask** keyword with the **network** command to specify individual subnets.
- Routes learned by the BGP process are propagated by default, but are often filtered by a routing policy.

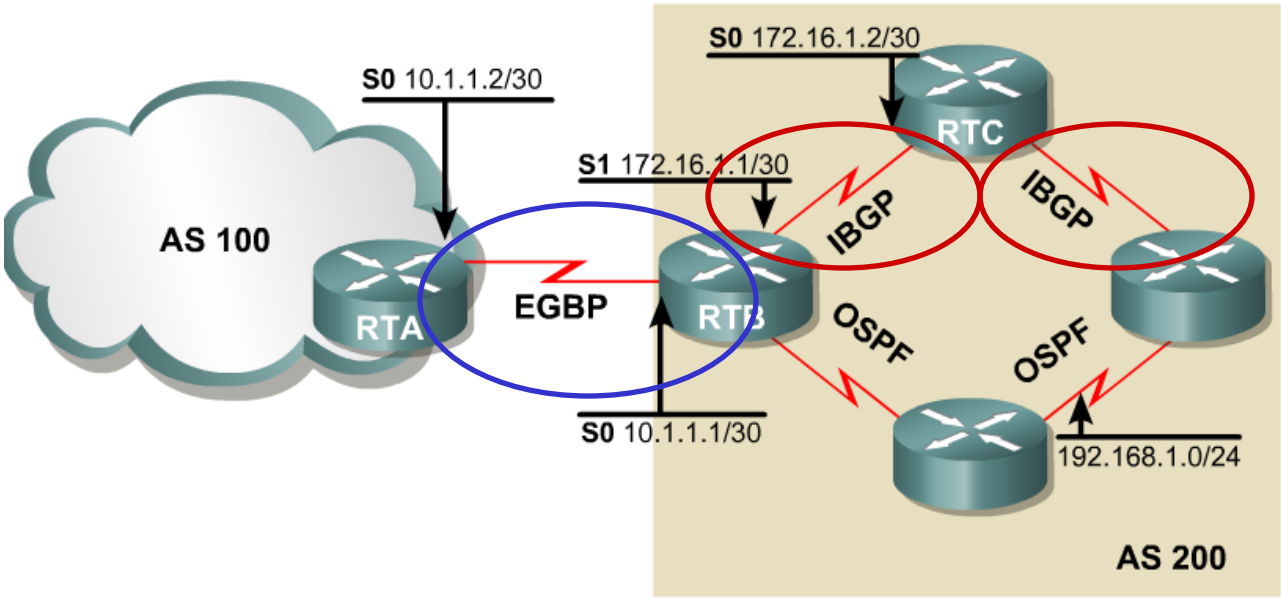
BGP Configuration



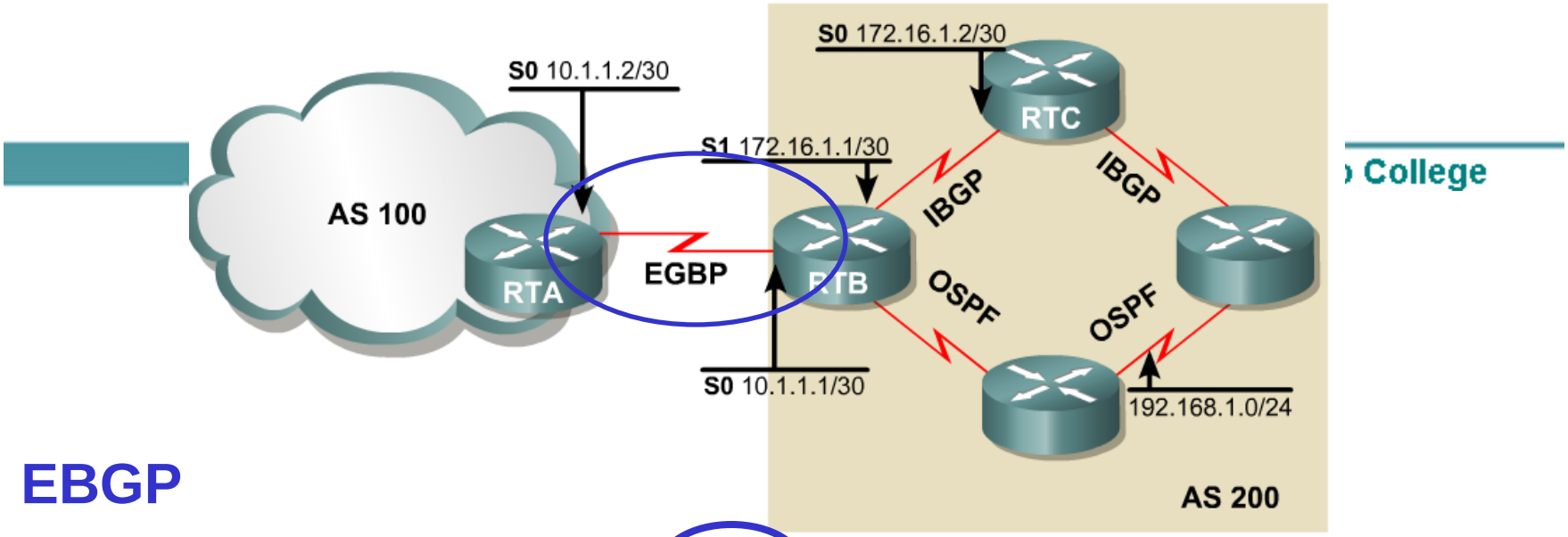
`Router(config-router)#`**`neighbor`** ***`ip-address`*** **`remote-as`** ***`AS-number`***

- In order for a BGP router to **establish a neighbor relationship with another BGP router**, you must issue the this configuration command.
- This command serves to identify a peer router with which the local router will establish a session.
- The ***AS-number*** argument determines whether the neighbor router is an EBGP or an IBGP neighbor.

BGP Configuration



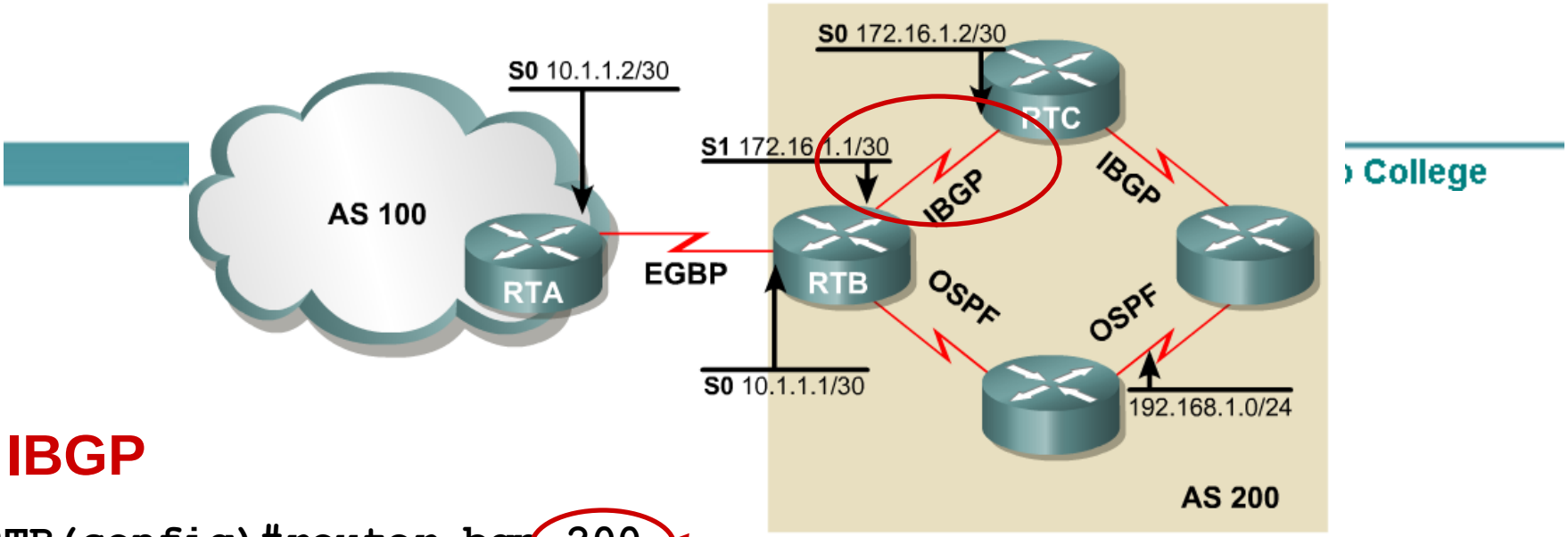
- If the ***AS-number*** configured in the **router bgp** command is **identical** to the ***AS-number*** configured in the **neighbor** statement, BGP will initiate an **internal session - IBGP**.
- If the field values are **different**, BGP will build an **external session - EBGP**.



EBGP

```
RTA(config)#router bgp 100
RTA(config-router)#neighbor 10.1.1.1 remote-as 200

RTB(config)#router bgp 200
RTB(config-router)#neighbor 10.1.1.2 remote-as 100
```

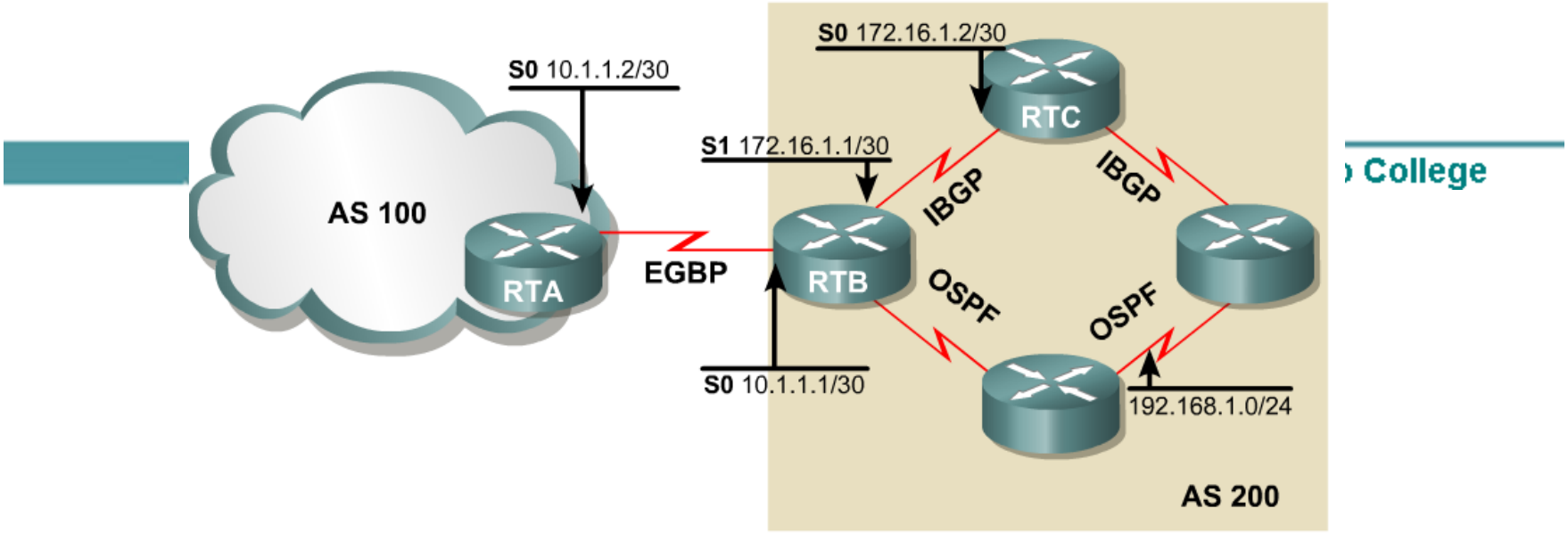
- RTB: Note that the **neighbor** command's **remote-as** value, 100, is different from the AS number specified by the **router bgp** command (200).
- Because the two AS numbers are different, BGP will start an **EBGP** connection with RTA.
- Communication will occur between autonomous systems.



IBGP

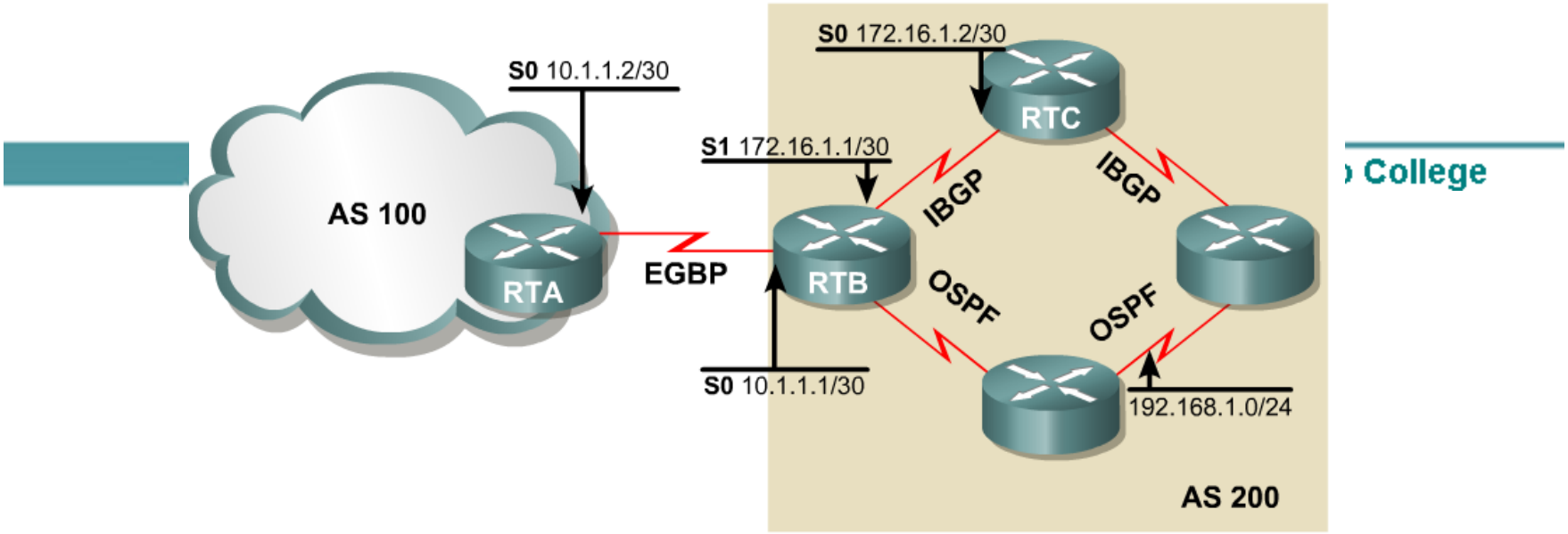
```
RTB(config)#router bgp 200
RTB(config-router)#neighbor 172.16.1.2 remote-as 200
RTB(config-router)#neighbor 172.16.1.2 update-source loopback 0

RTC(config)#router bgp 200
RTC(config-router)#neighbor 172.16.1.1 remote-as 200
RTC(config-router)#neighbor 172.16.1.1 update-source loopback 0
```

- Since the **remote-as** value (200) is the same as RTB's BGP AS number, BGP recognizes that this connection will occur within AS 200, so it attempts to establish an IBGP session.
- In reality, AS 200 is not a remote AS at all; it is the local AS, since both routers live there. But for simplicity, the keyword **remote-as** is used when configuring both EBGP and IBGP sessions.




```
RTB(config-router)#neighbor 172.16.1.2 update-source loopback 0
RTC(config-router)#neighbor 172.16.1.1 update-source loopback 0
```

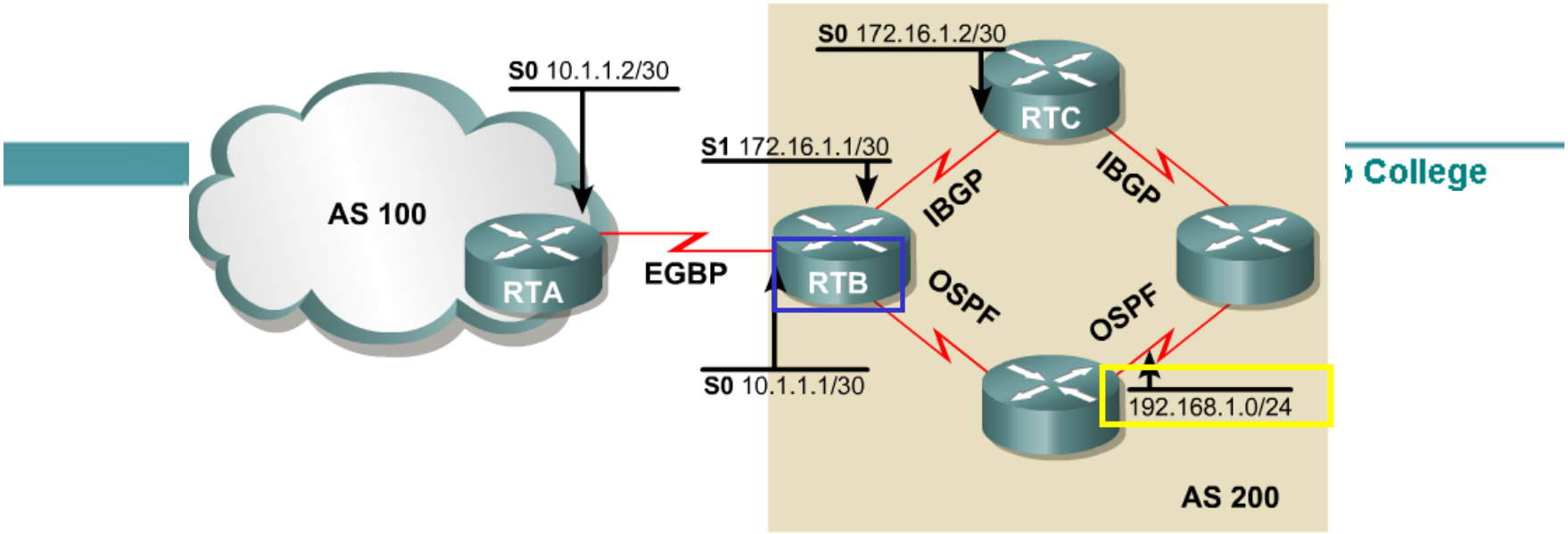
- The **update-source loopback 0** command is used to instruct the router to use *any* operational interface for TCP connections (as long as Lo0 is up and configured with an IP address).
- Without the **update-source loopback 0** command, BGP routers can use only the closest IP interface to the peer.
- The ability to use any operational interface provides BGP with robustness in the event the link to the closet interface fails.
  - Since EBGP sessions are typically point-to-point, there is no need to use this command with EBGP.



- Assume the following route appears in RTB's table:

```
0 192.168.1.0/24 [110/74] via 10.2.2.1, 00:31:34,
  Serial2
```

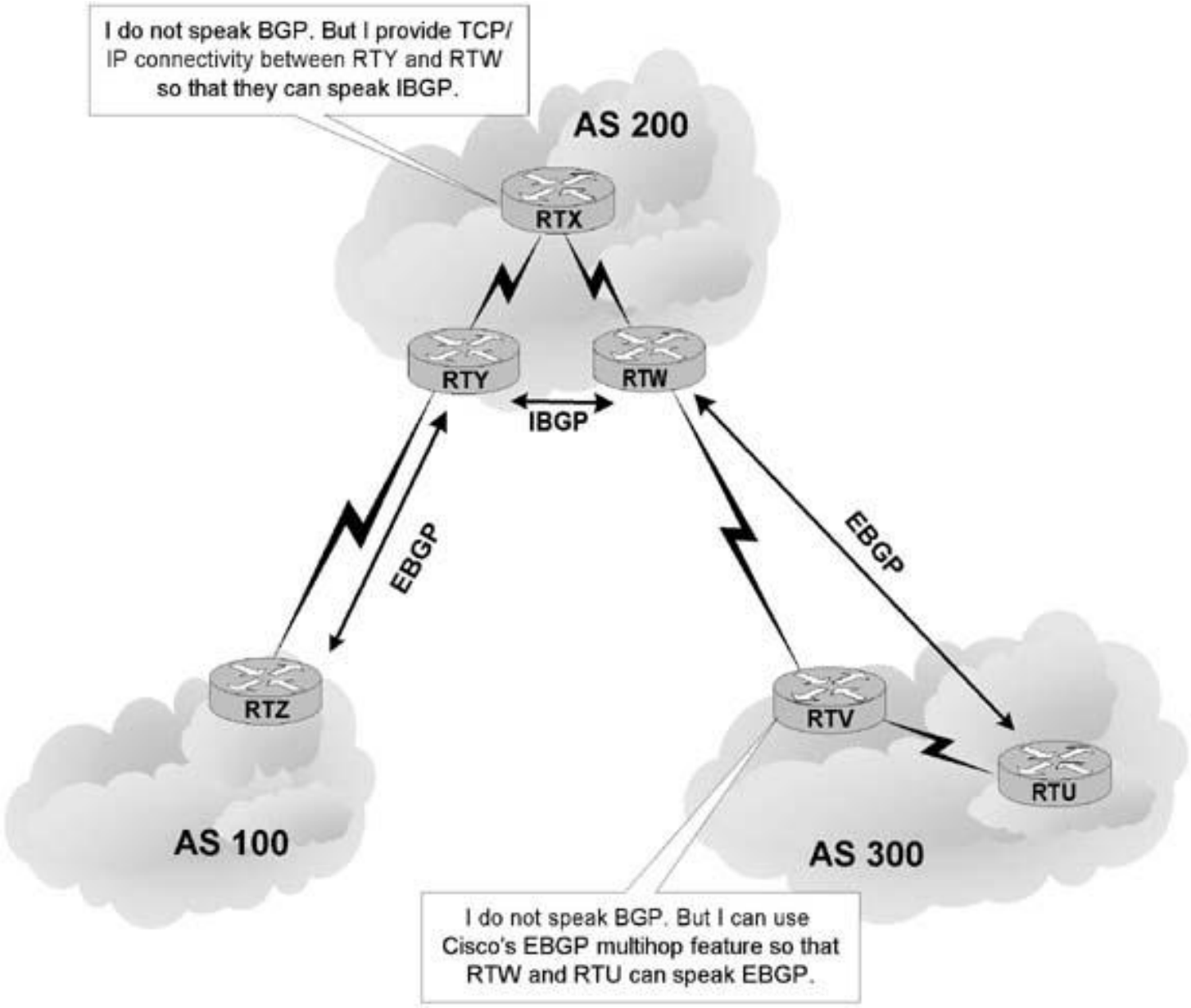
- RTB learned this route via an IGP, in this case, OSPF.
- This AS uses OSPF internally to exchange route information.
- Can RTB advertise this network via BGP?
- Certainly, redistributing OSPF into BGP will do the trick, but the BGP **network** command will do the same thing.



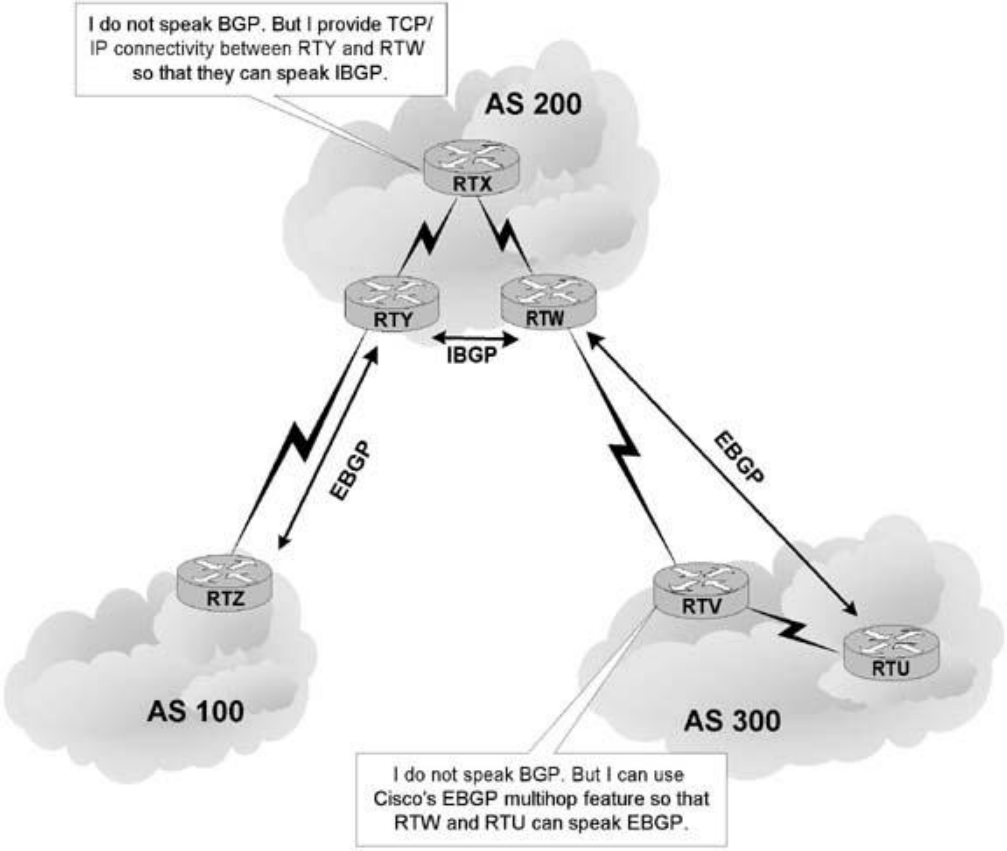

```
RTB(config)#router bgp 200
RTB(config-router)#network 172.16.1.0 mask 255.255.255.254
RTB(config-router)#network 10.1.1.0 mask 255.255.255.254
RTB(config-router)#network 192.168.1.0
```

- The first two **network** commands in include the **mask** keyword, so that only a particular subnet is specified.
- The third **network** command results in the OSPF route being advertised by BGP *without* redistribution.
- Remember that the BGP **network** command works differently than the IGP **network** command!

EBGP vs IBGP

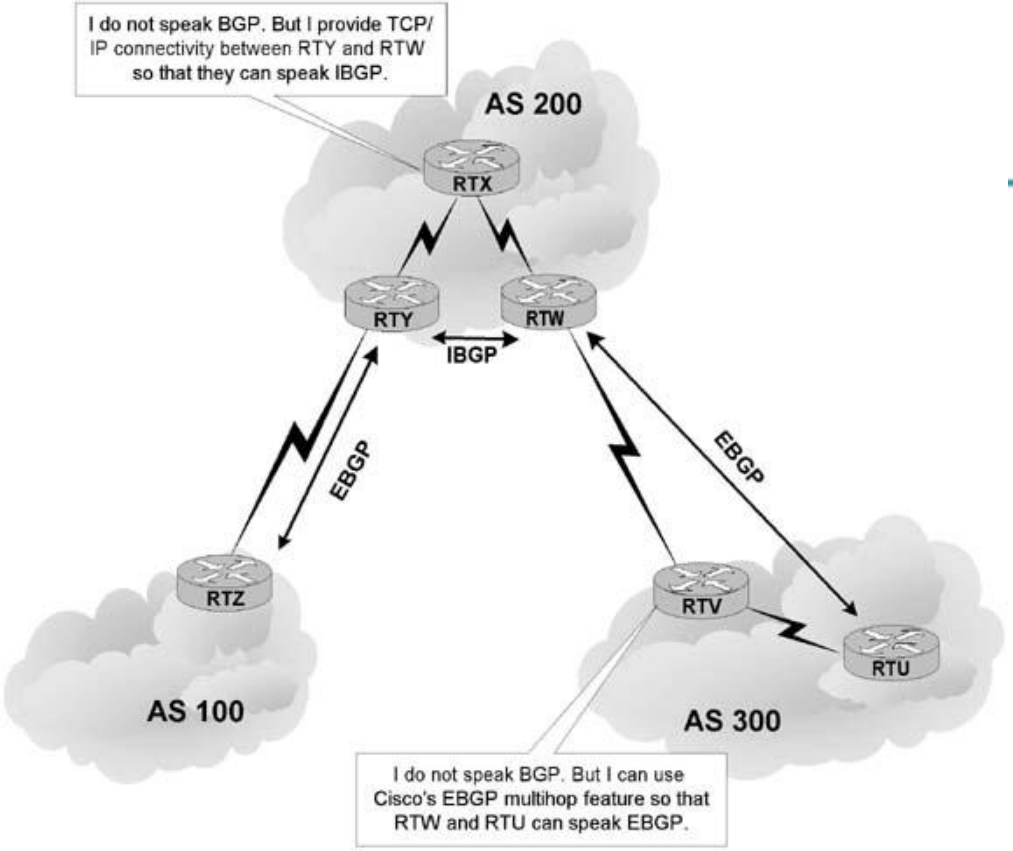

I do not speak BGP. But I provide TCP/IP connectivity between RTY and RTW so that they can speak IBGP.
AS 200
RTX
RTY
RTW
IBGP
EBGP
EBGP
RTZ
RTV
RTU
AS 100
AS 300
I do not speak BGP. But I can use Cisco's EBGP multihop feature so that RTW and RTU can speak EBGP.

EBGP vs IBGP



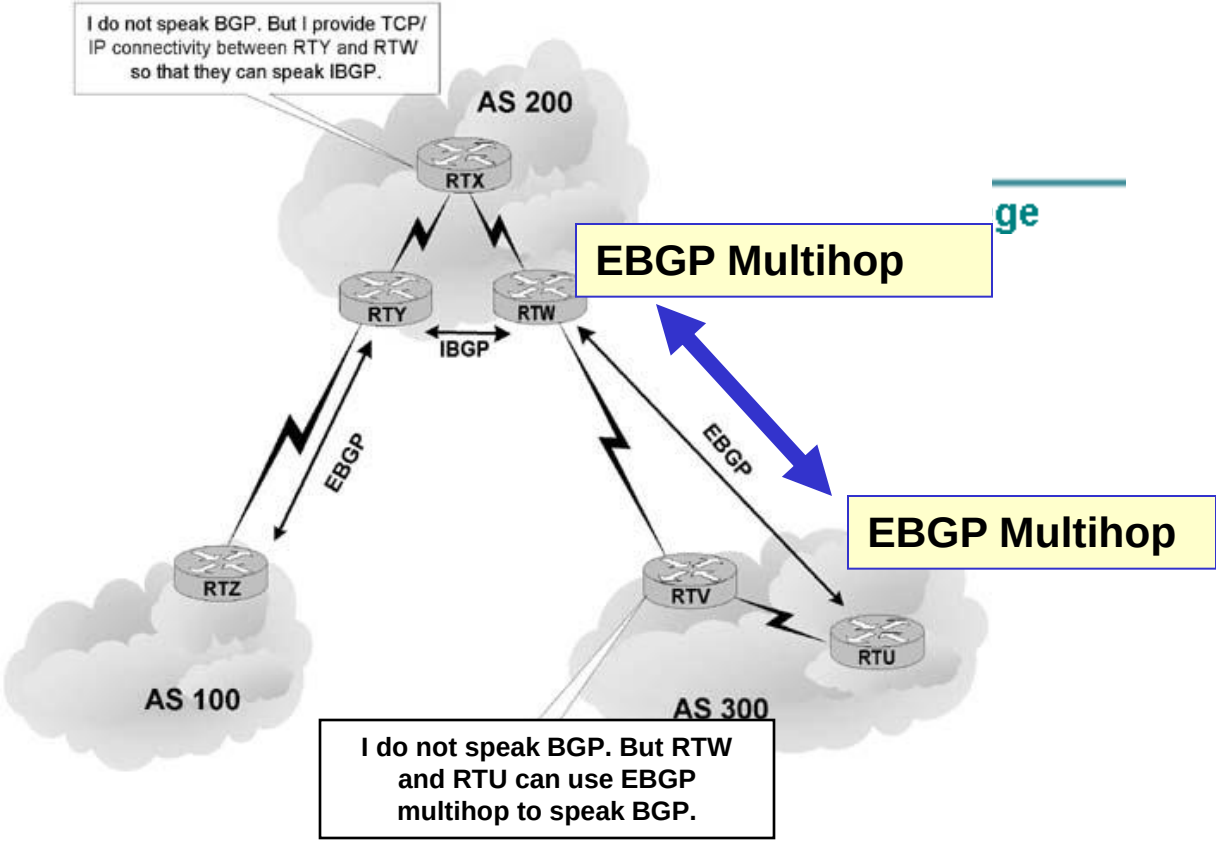
- **EBGP** peers must be **directly connected**, but there are certain exceptions to this requirement.
- In contrast, **IBGP** peers merely **require TCP/IP connectivity** within the same AS.
  - As long as RTY can communicate with RTW using TCP, both routers can establish an IBGP session.
  - If needed, an IGP such as OSPF can provide IBGP peers with routes to each other.

IBGP



- In a typical configuration, an **IBGP** router maintains **IBGP** sessions with all other **IBGP** routers in the AS, forming a logical full-mesh.
  - This is necessary because IBGP routers do not advertise routes learned via IBGP to other IBGP peers (to prevent routing loops).
  - In other words, if you want your IBGP routers to exchange BGP routes with each other, you should configure a full-mesh.
  - An alternative to this approach: configuring a route reflector (later)

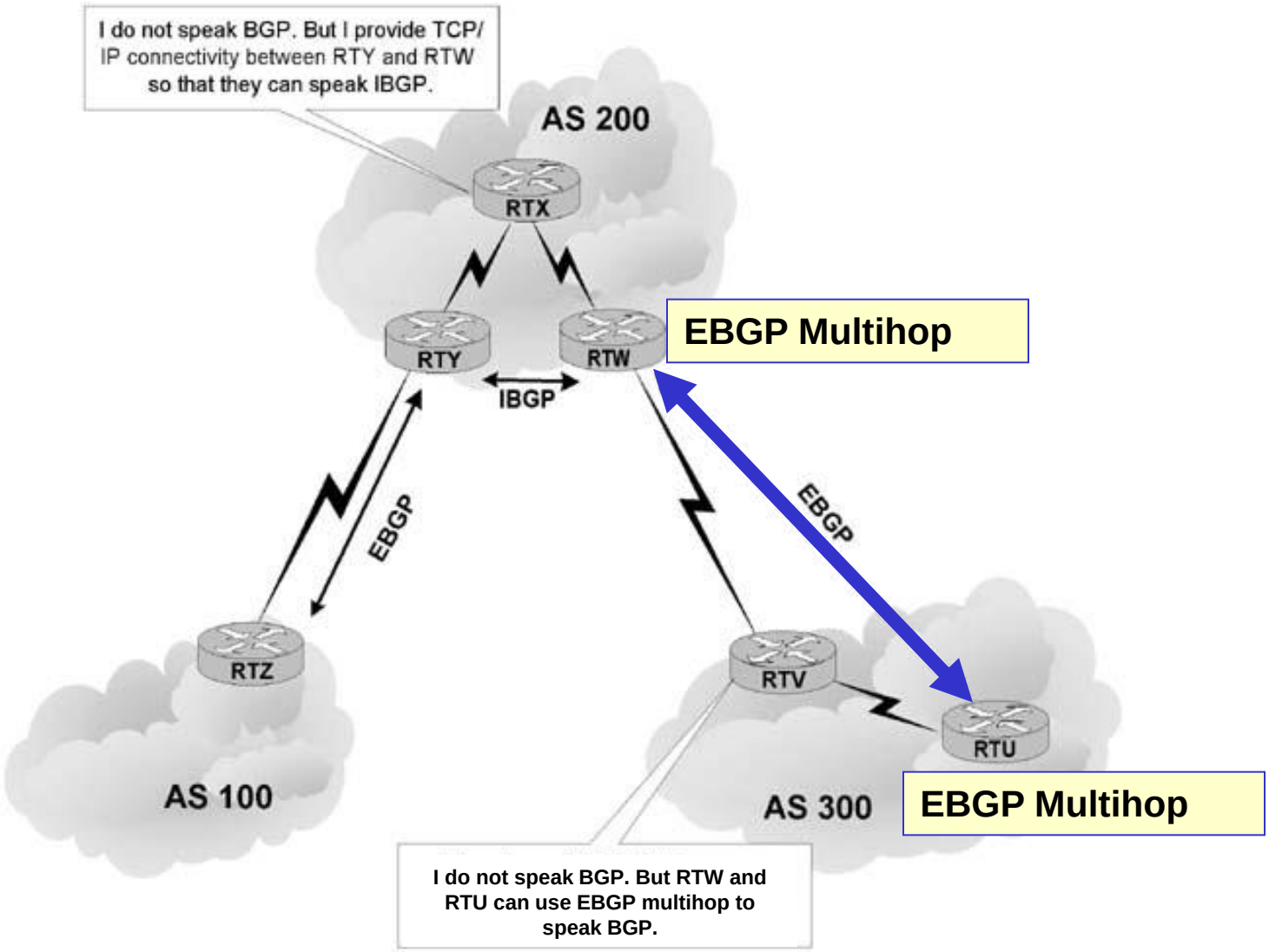
EBGP



- **EBGP** neighbors must be directly connected in order to establish an **EBGP** session.
- However, **EBGP** **multihop** is a Cisco IOS option allows RTW and RTU to be logically connected in an **EBGP** session, despite the fact that RTV does not support BGP.
- The **EBGP** multihop option is configured on each peer with the following command:

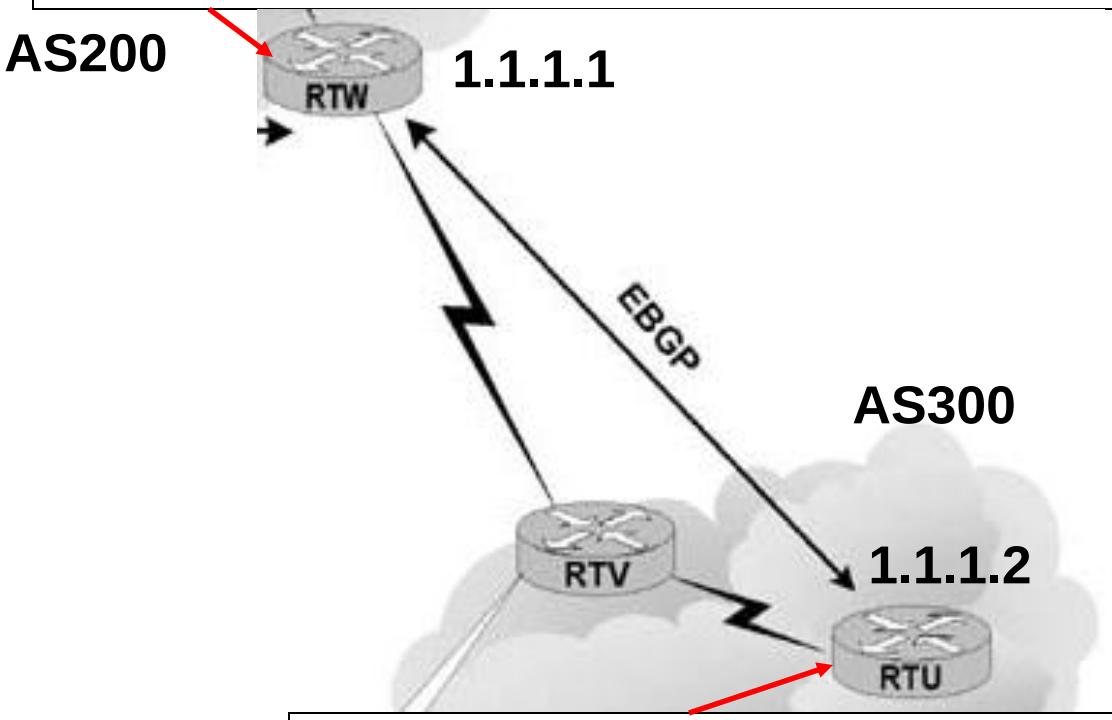
```
Router(config-router)#neighbor IP-address ebgp-multihop [hops]
```

EBGP


I do not speak BGP. But I provide TCP/IP connectivity between RTY and RTW so that they can speak IBGP.
AS 200
RTX
RTY
RTW
IBGP
EBGP Multihop
EBGP
EBGP
RTZ
RTV
RTU
AS 100
AS 300
EBGP Multihop
I do not speak BGP. But RTW and RTU can use EBGP multihop to speak BGP.

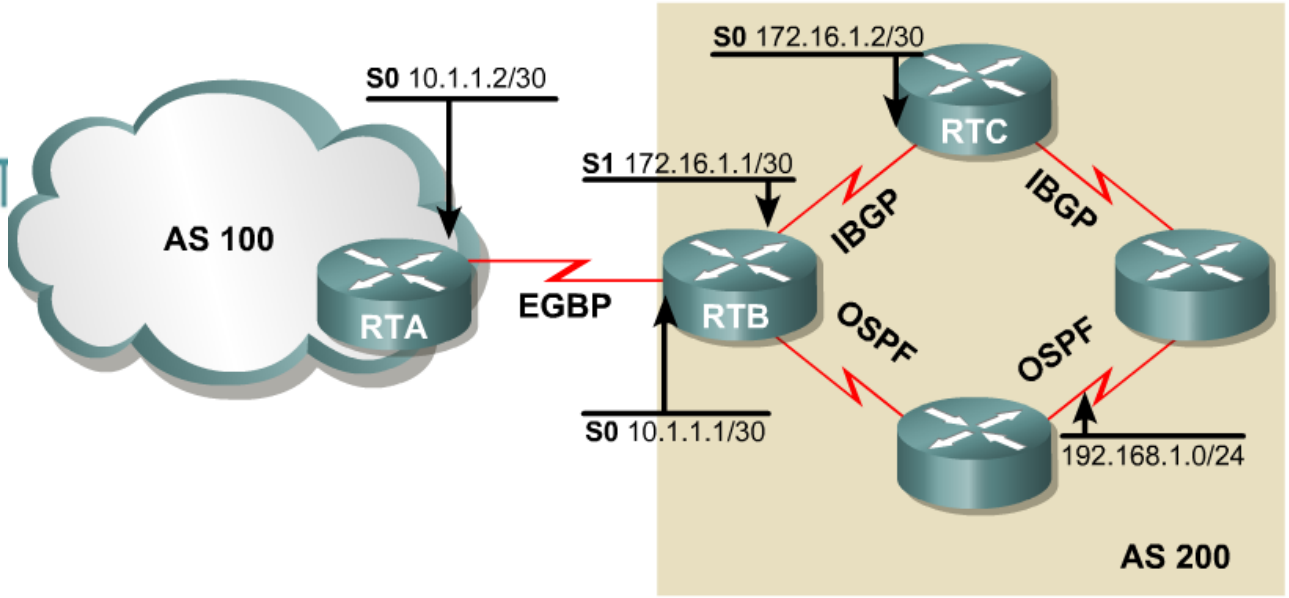
EBGP Multihop


```
RTW(config)#router bgp 200
RTW(config-router)#neighbor 1.1.1.2 remote-as 300
RTW(config-router)#neighbor 1.1.1.2 ebgp-multihop 2
```




```
RTU(config)#router bgp 300
RTU(config-router)#neighbor 1.1.1.1 remote-as 200
RTU(config-router)#neighbor 1.1.1.1 ebgp-multihop 2
```

BGP Configuration



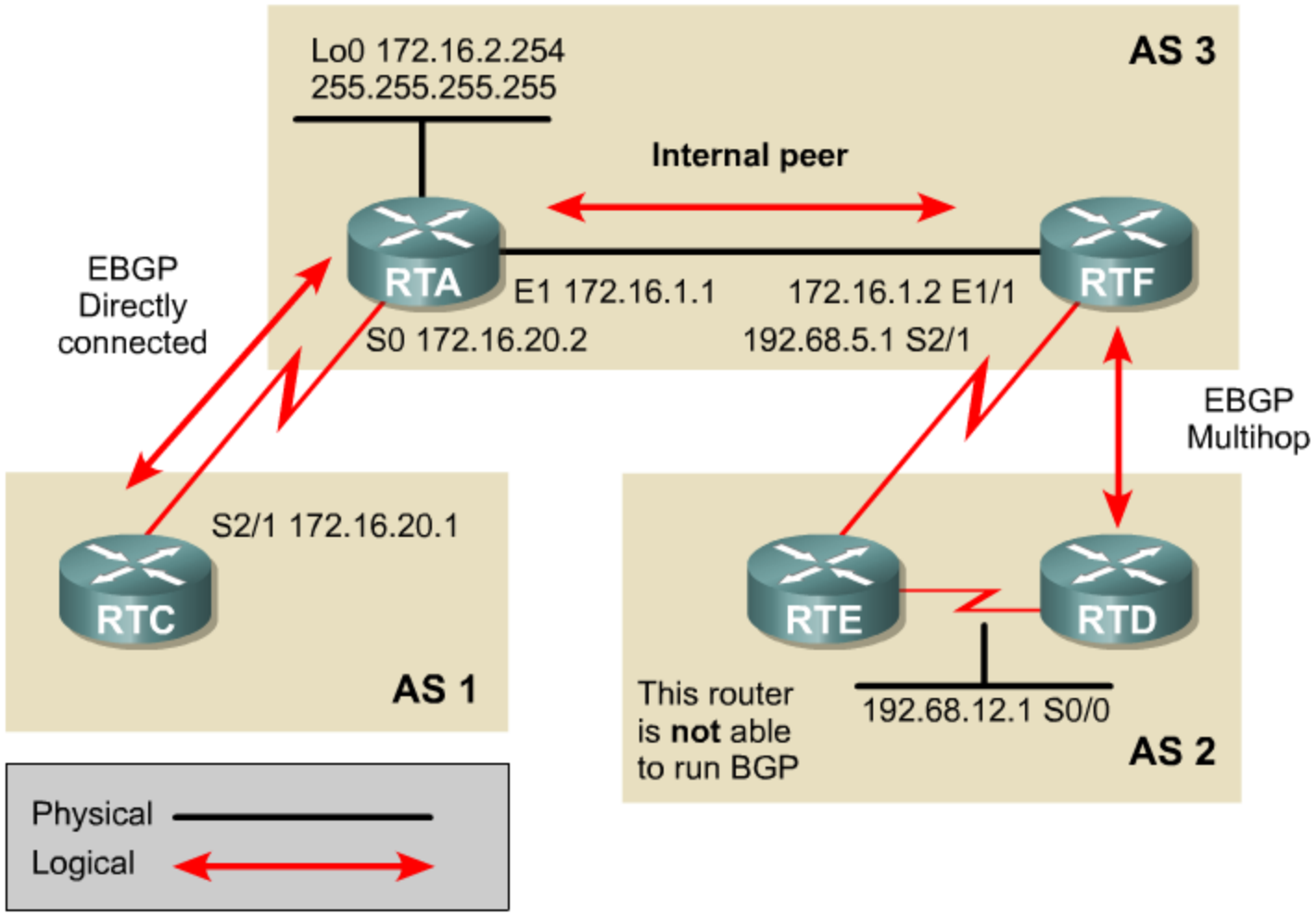
- Finally, whenever you are configuring BGP, you will notice that changes you make to an existing configuration may not appear immediately.
- To force BGP to clear its table and reset BGP sessions, use the **clear ip bgp** command. The easiest way to enter this command is as follows:

```
Router#clear ip bgp *
Router#clear ip bgp 10.0.0.0
```

Use this command with CAUTION, better yet, not at all, in a production network. From the net…

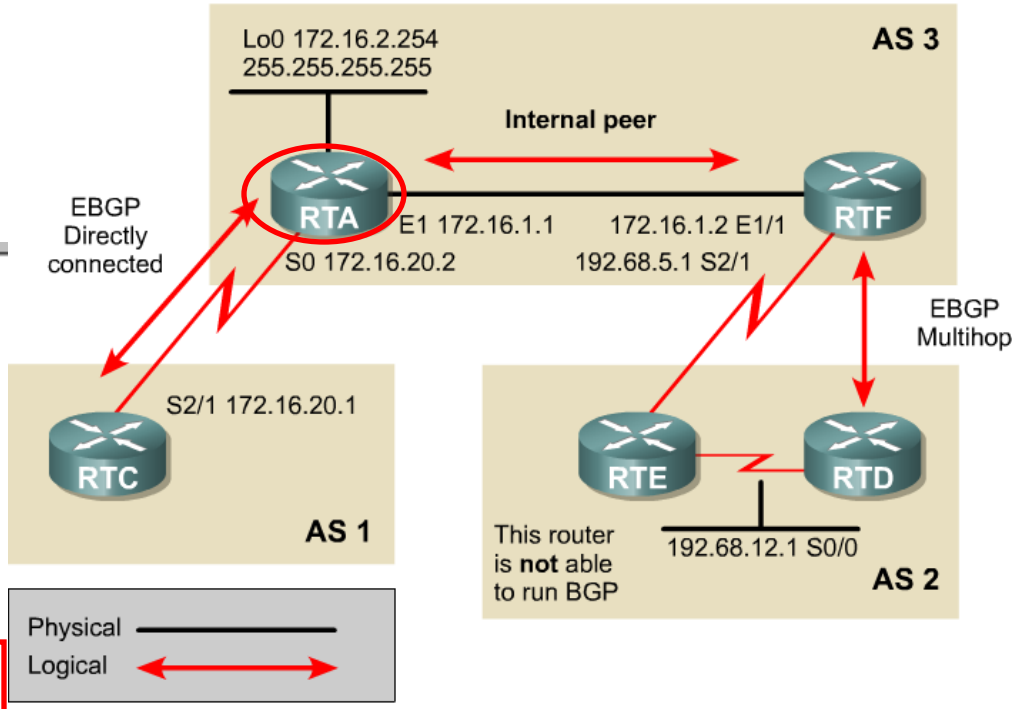
- "clear ip bgp * OOPS! Not me but a colleague who was an employee of a large ISP with a 3 letter title. Got back from a Cisco routing course and thought they would try out some commands on the core network. It took 45 minutes for the core to reconverge. P45 followed"

Example


Lo0 172.16.2.254
255.255.255.255
AS 3
Internal peer
EBGP
Directly
connected
RTA
E1 172.16.1.1
172.16.1.2 E1/1
RTF
S0 172.16.20.2
192.68.5.1 S2/1
EBGP
Multihop
S2/1 172.16.20.1
RTC
AS 1
RTE
RTD
This router
is not able
to run BGP
192.68.12.1 S0/0
AS 2
Physical
Logical

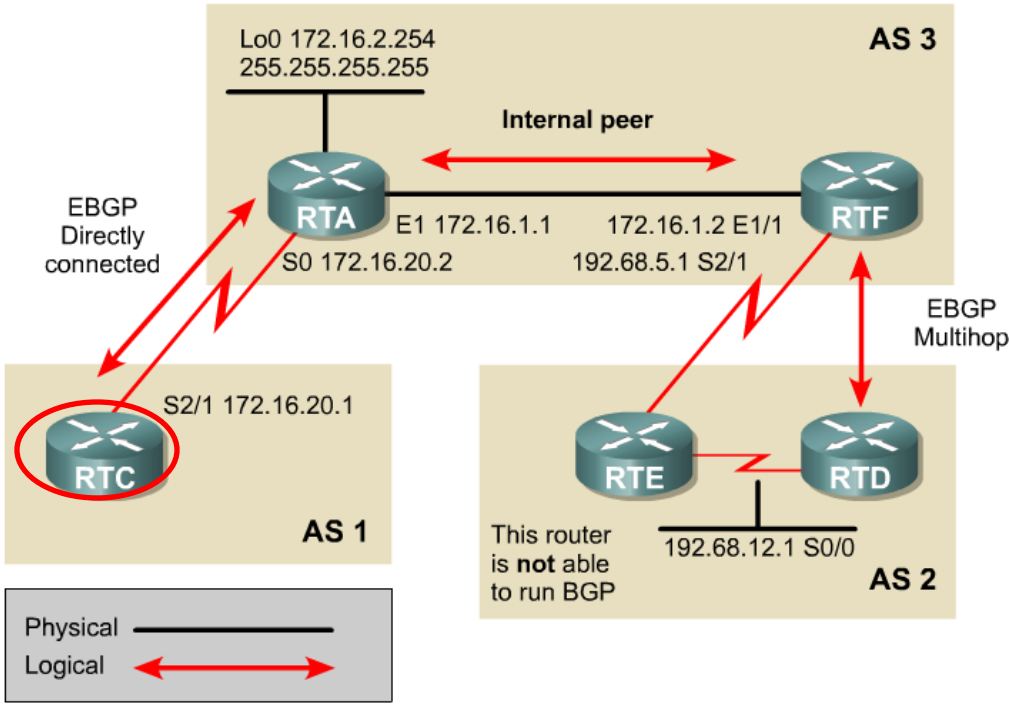
Example

```
RTA#show running-config
ip subnet-zero
interface Loopback0
ip address 172.16.2.254 255.255.255.255
interface Ethernet1
ip address 172.16.1.1 255.255.255.0
interface Serial0
ip address 172.16.20.2 255.255.255.0
router ospf 10
network 172.16.0.0 0.0.255.255 area 0
router bgp 3
no synchronization
neighbor 172.16.1.2 remote-as 3
neighbor 172.16.1.2 update-source Loopback0
neighbor 172.16.20.1 remote-as 1
no auto-summary
ip classless
RTA#
```



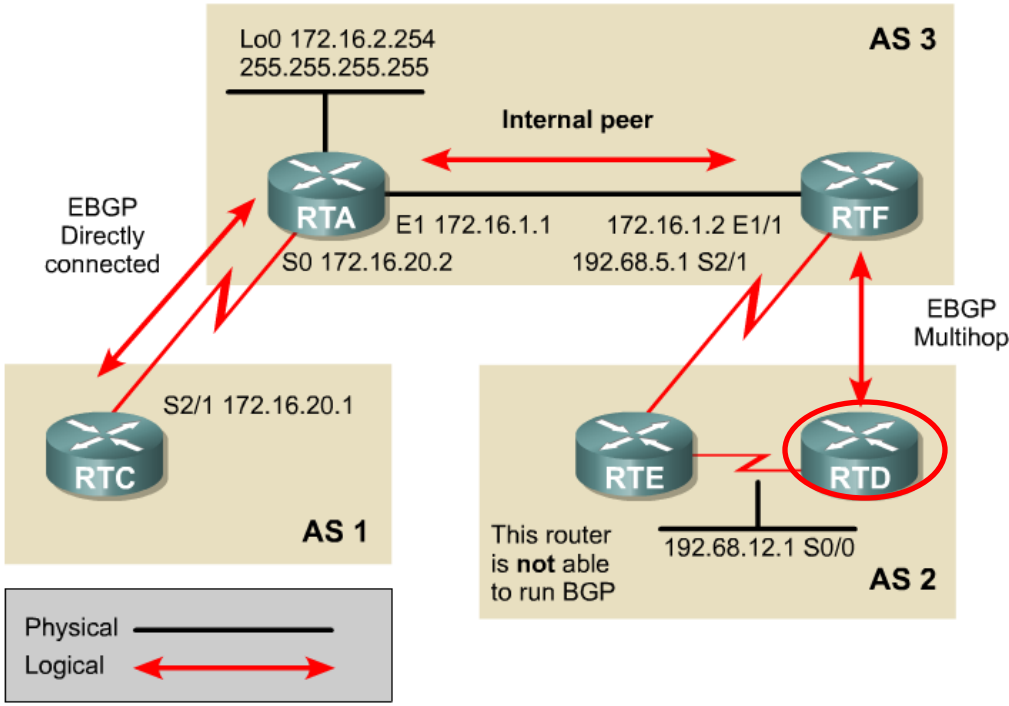
Example

```
RTC#show running-config
ip subnet-zero
interface Serial2/1
ip address 172.16.20.1 255.255.255.0
router bgp 1
neighbor 172.16.20.2 remote-as 3
no auto-summary
ip classless
RTC#
```



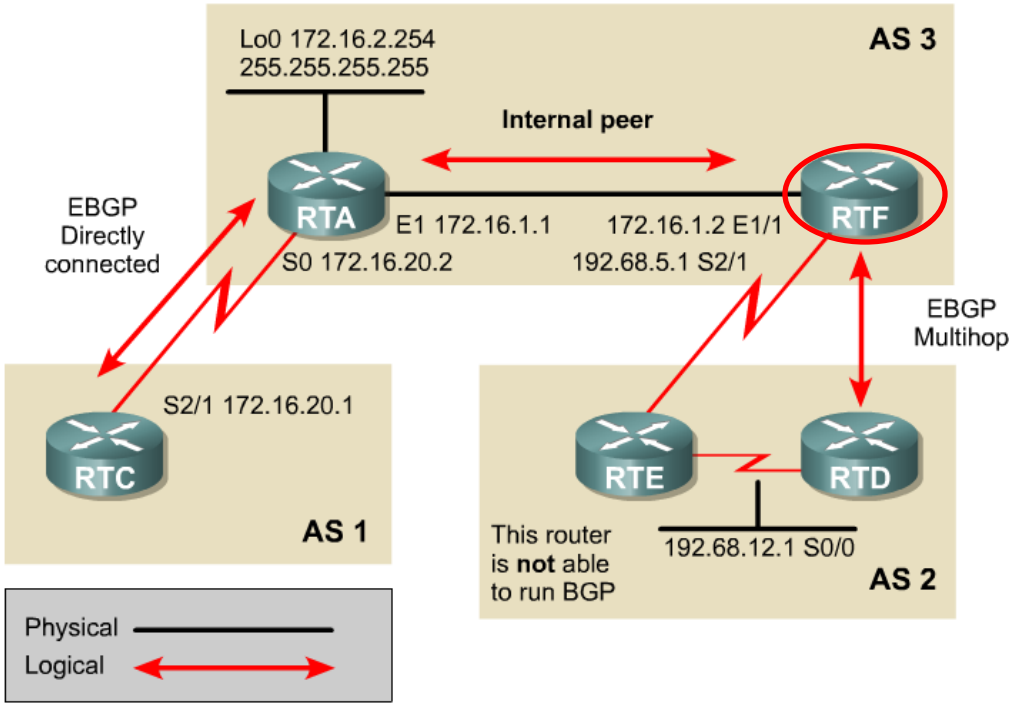
Example

```
RTD#show running-config
ip subnet-zero
interface Serial0/0
ip address 192.68.12.1 255.255.255.0
router ospf 10
network 192.68.0.0 0.0.255.255 area 0
router bgp 2
neighbor 192.68.5.1 remote-as 3
neighbor 192.68.5.1 ebgp-multihop 2
no auto-summary
ip classless
RTD#
```



Example

```
RTF#show running-config
ip subnet-zero
interface Ethernet1/1
ip address 172.16.1.2 255.255.255.0
interface Serial2/1
ip address 192.68.5.1 255.255.255.0
router ospf 10
network 172.16.0.0 0.0.255.255 area 0
network 192.68.0.0 0.0.255.255 area 0
router bgp 3
no synchronization
neighbor 172.16.2.254 remote-as 3
neighbor 192.68.12.1 remote-as 2
neighbor 192.68.12.1 ebgp-multihop 2
no auto-summary
ip classless
RTF#
```




```
RTF#show ip bgp neighbor
BGP neighbor is 172.16.2.254, remote AS 3, internal link
BGP version 4, remote router ID 172.16.2.254
BGP state = Established, table version = 2, up for 22:36:09
Last read 00:00:10, hold time is 180, keepalive interval is 60 seconds
Minimum time between advertisement runs is 5 seconds
Received 1362 messages, 0 notifications, 0 in queue
Sent 1362 messages, 0 notifications, 0 in queue
Connections established 2; dropped 1
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 172.16.1.2, Local port: 11008
Foreign host: 172.16.2.254, Foreign port: 179
BGP neighbor is 192.68.12.1, remote AS 2, external link
BGP version 4, remote router ID 192.68.5.2
BGP state = Established, table version = 2, up for 22:13:01
Last read 00:00:00, hold time is 180, keepalive interval is 60 seconds
Minimum time between advertisement runs is 30 seconds
Received 1336 messages, 0 notifications, 0 in queue
```

Verifying BGP Configuration

- If the router has not installed the BGP routes you expect, you can use the **show ip bgp** command to verify that BGP has learned these routes.
- ***More later…***

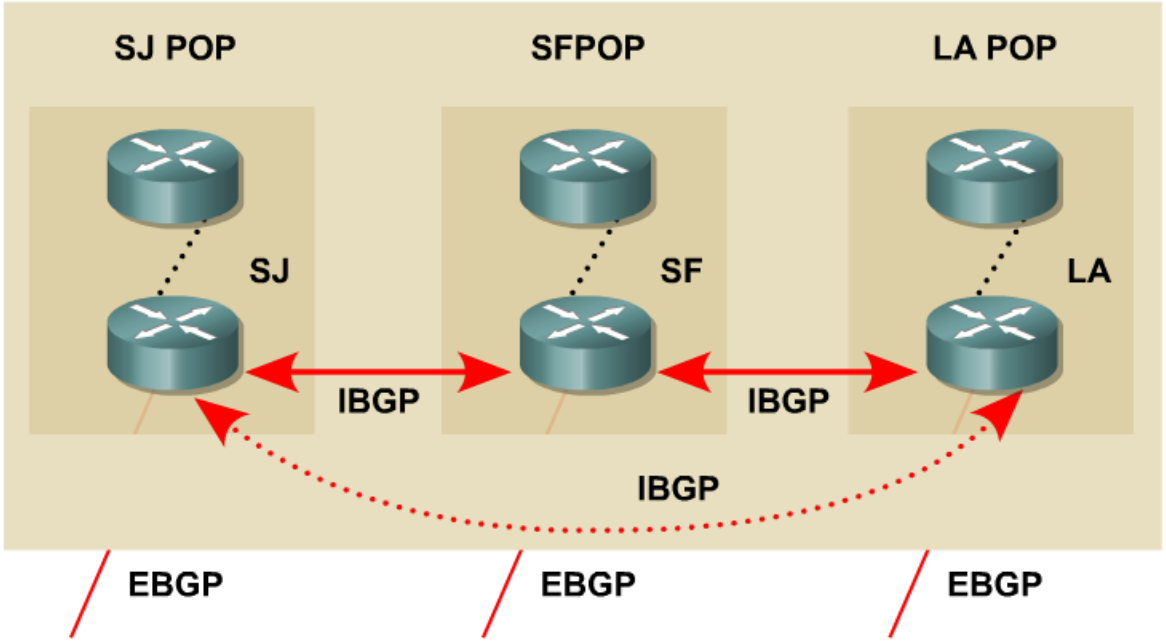
```
RTA#show ip bgp
BGP table version is 3, local router ID is 10.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i1.0.0.0         192.168.1.6              0    100      0 200 400 e
*>i10.1.1.1/32     10.1.1.1                 0    100      0 i
*>i172.16.1.0/24   10.1.1.1                 0    100      0 i
* i192.168.1.32/27 192.168.1.6              0    100      0 200 i
```

Verifying BGP Configuration

- If an expected BGP route does not appear in the BGP table, you can use the **show ip bgp neighbors** command to verify that your router has established a BGP connection with its neighbors.

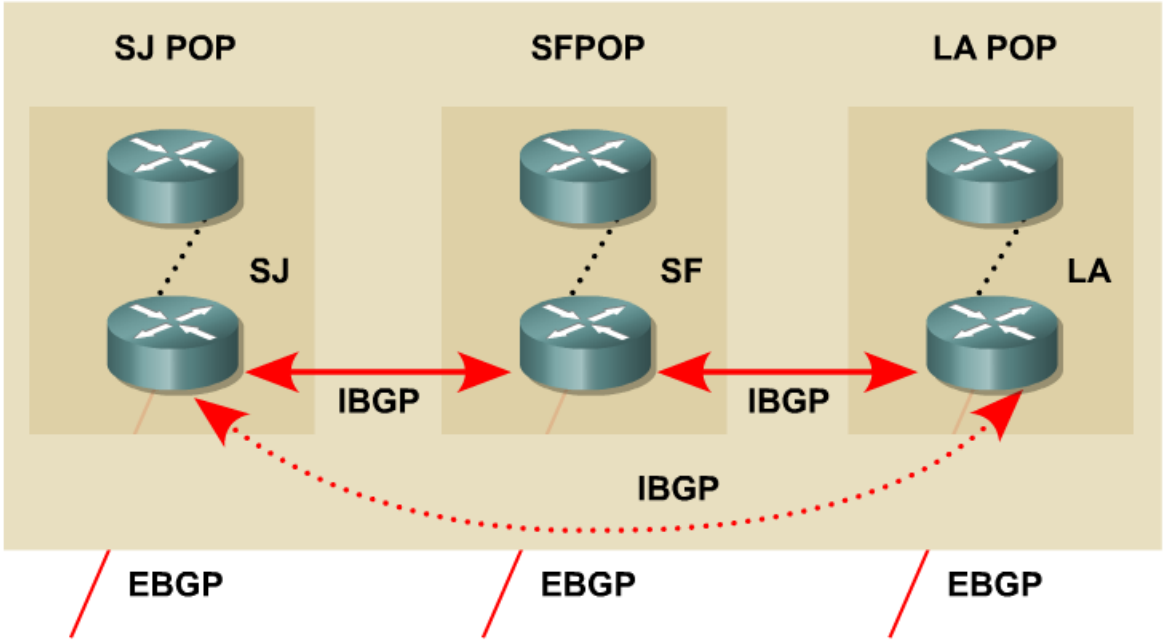
```
RTA#show ip bgp neighbors
BGP neighbor is 172.24.1.18,  remote AS 200, external link
  BGP version 4, remote router ID 172.16.1.1
  BGP state = Established, up for 00:03:25
  Last read 00:00:25, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received
    Address family IPv4 Unicast: advertised and received
  Received 7 messages, 0 notifications, 0 in queue
  Sent 8 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 30 seconds
<output omitted>
```

BGP Peering



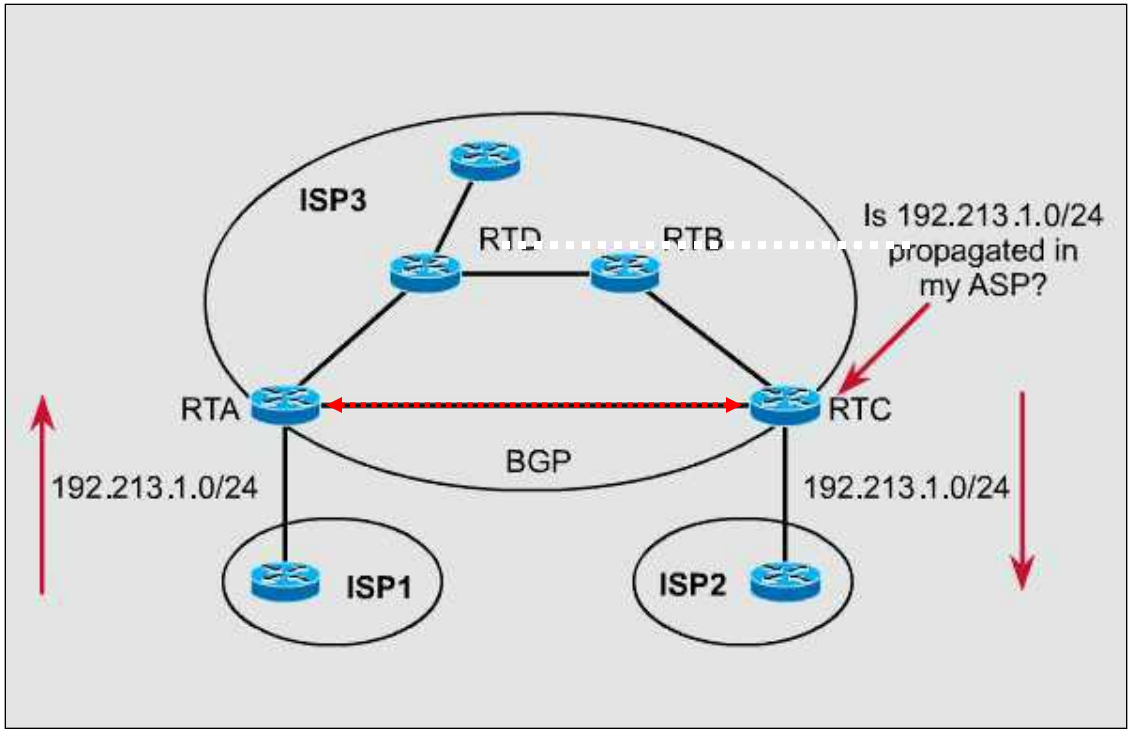
- Routes learned via IBGP peers are **not** propagated to other IBGP peers.
  – **BGP Split Horizon Rule**
- If they did, BGP routing inside the AS would present a dangerous potential for routing loops.
- For IBGP routers to learn about all BGP routes inside the AS, they must connect to every other IBGP router in a logical full IBGP mesh.
  – You can create a logical full mesh even if the routers aren't directly connected, as long as the IBGP peers can connect to each other using TCP/IP.

BGP Peering



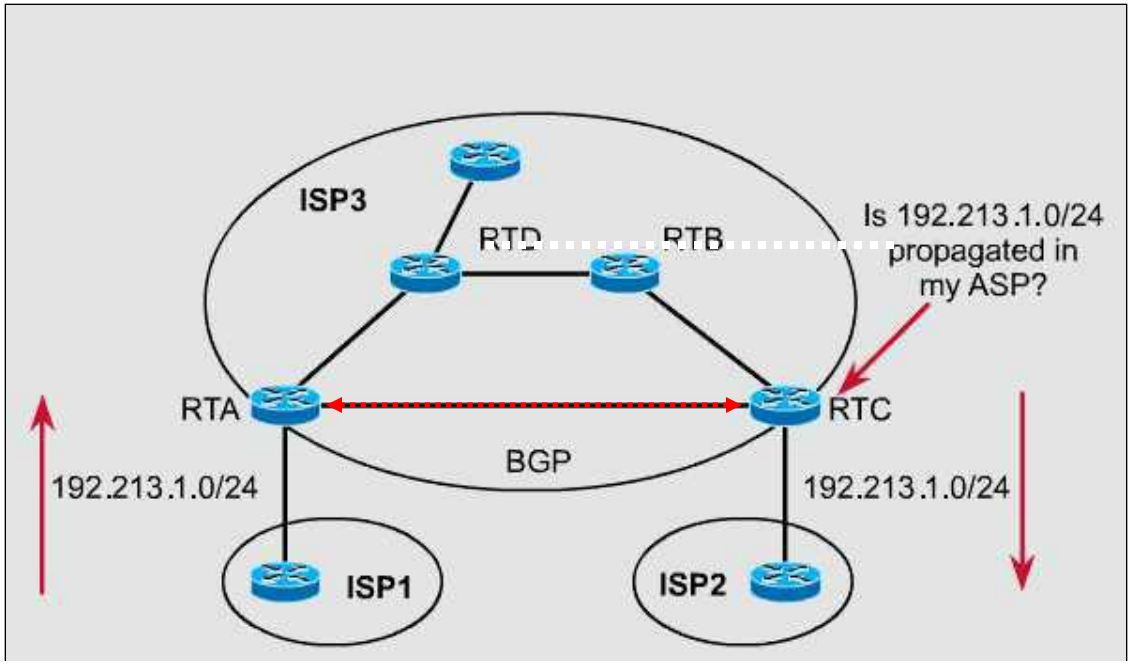
- Without dotted connection, routing in this scenario is not complete.
- EBGP routes learned by way of San Jose will not be given to Los Angeles, and EBGP routes learned by way of Los Angeles will not be given to San Jose.
- This is because the San Francisco router will not advertise IBGP routes between San Jose and Los Angeles.
- What is needed is an additional IBGP connection between San Jose and Los Angeles.
- This connection is shown as a dotted line.

AS Synchronization



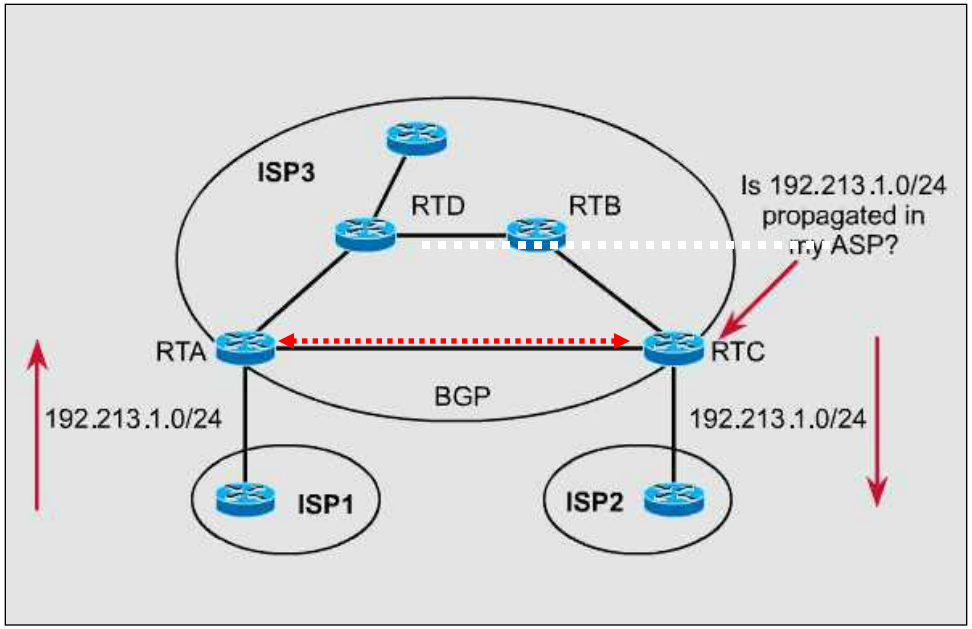
- When an IBGP router receives an update about a destination from an IBGP peer, it tries to verify reachability to that destination via an **IGP**, such as RIP or OSPF.
- If the IBGP router can't find the destination network in it's **IGP** routing table, it **will not** advertise the destination to other BGP peers.

AS Synchronization



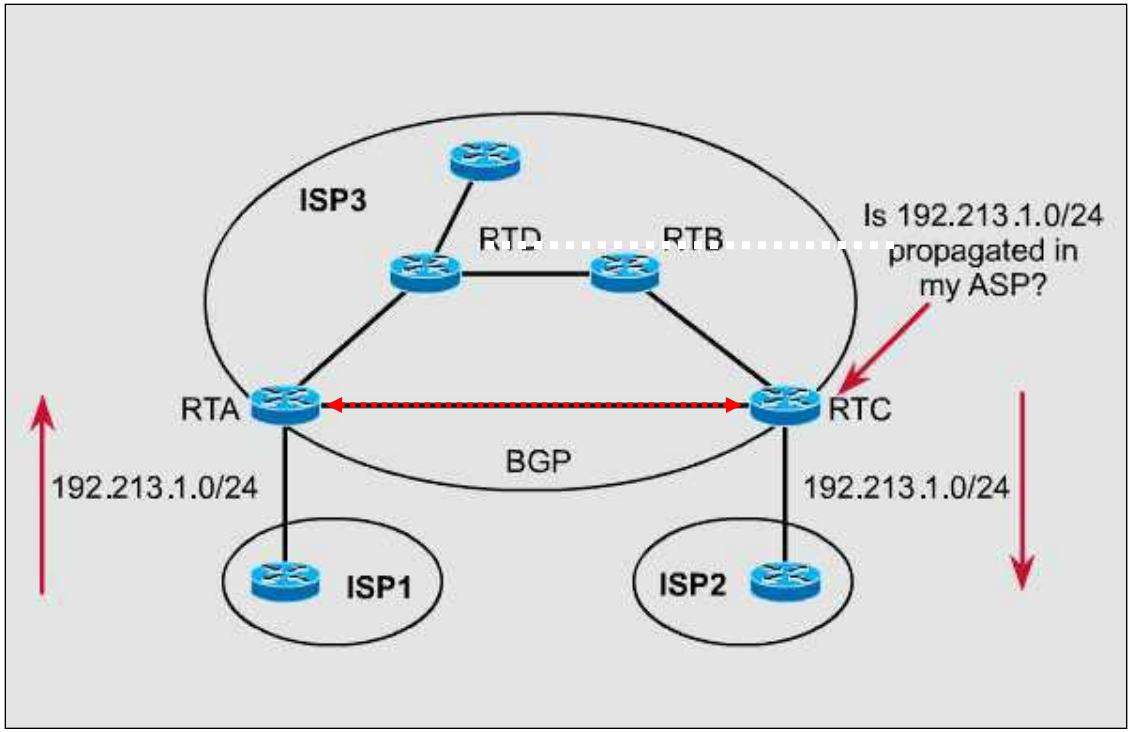
- If the route is **not** reachable through the **IGP** running within the AS, non-BGP routers will not be able to route traffic passing through the AS towards this destination.
- It is pointless to advertise destinations to external peers if traffic sent through this AS is going to be dropped by some non-BGP router within the AS anyway.

AS Synchronization



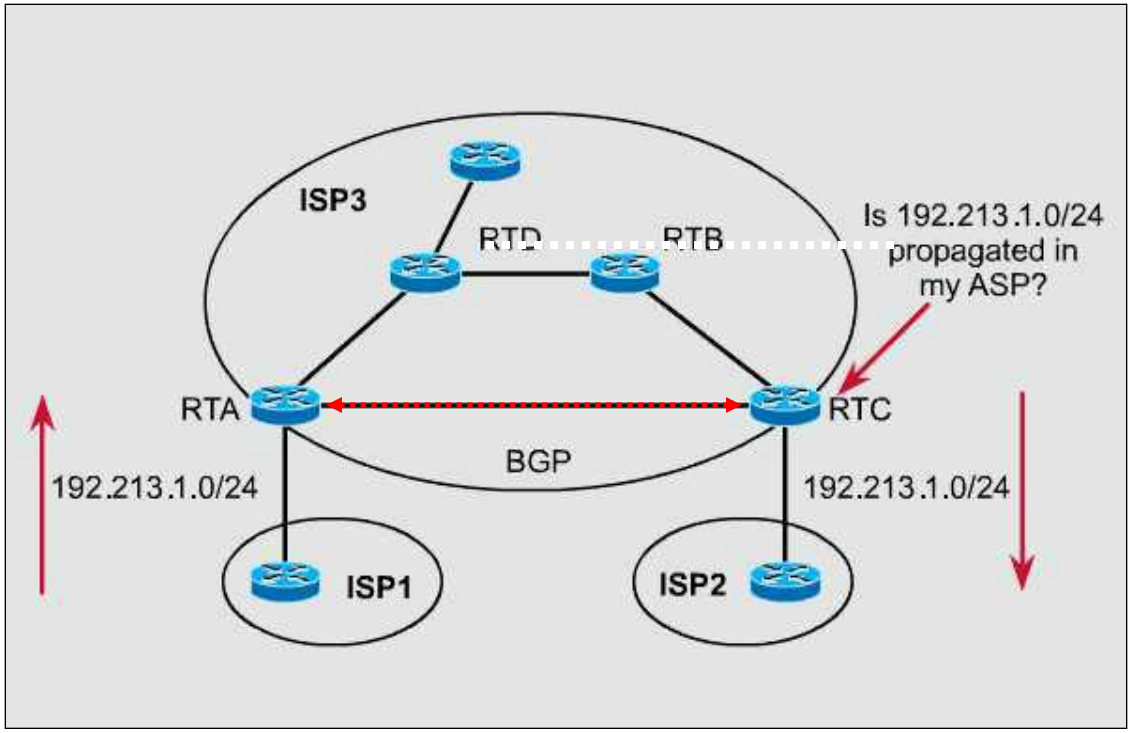
- The **BGP synchronization rule** states that a BGP router **(RTC)** should not advertise to external neighbors **(ISP2)** destinations **(192.213.1.0/24)** learned from inside BGP neighbors **(RTA)** unless those destinations are also known via an **IGP** **(RTD and RTB)**.
- If a router knows about these destinations via an **IGP**, it assumes that the route has already been propagated inside the AS, and internal reachability is guaranteed.

AS Synchronization



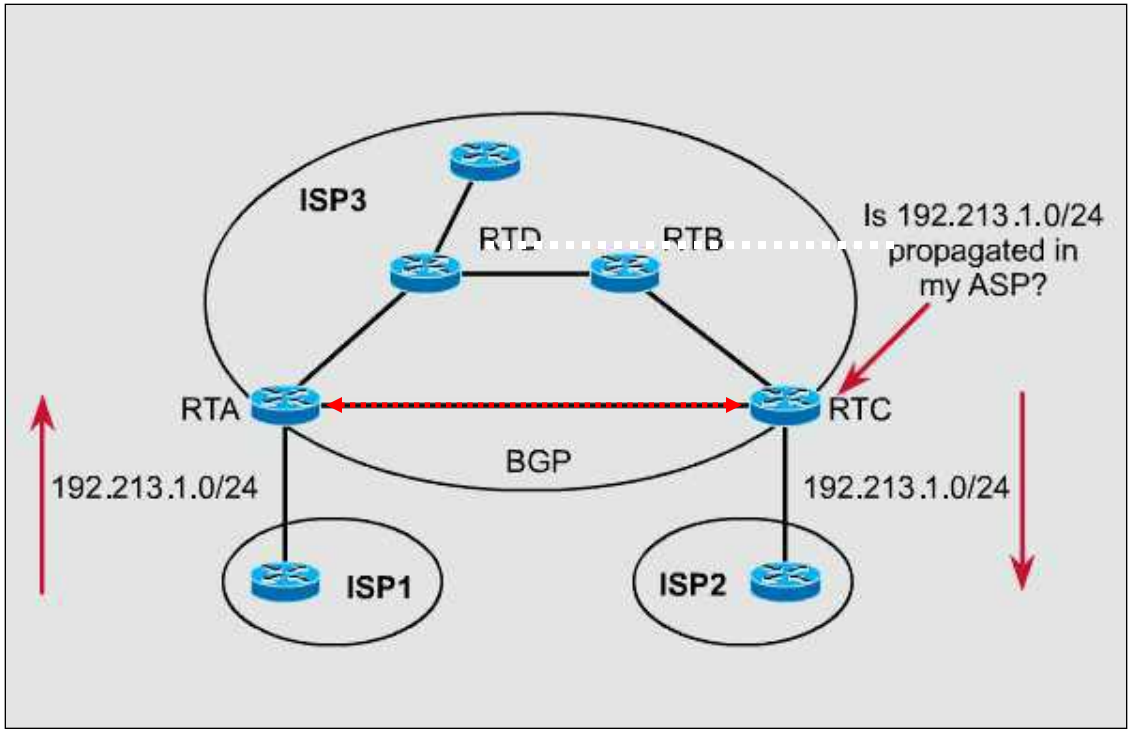
- If the IBGP router **(RTC)** does have an **IGP** route to this destination, the route is considered ***synchronized***, and the router will announce it to other BGP peers **(ISP2).**
- *Otherwise*, the router will treat the route as not being synchronized with the **IGP** and will not advertise it.

AS Synchronization



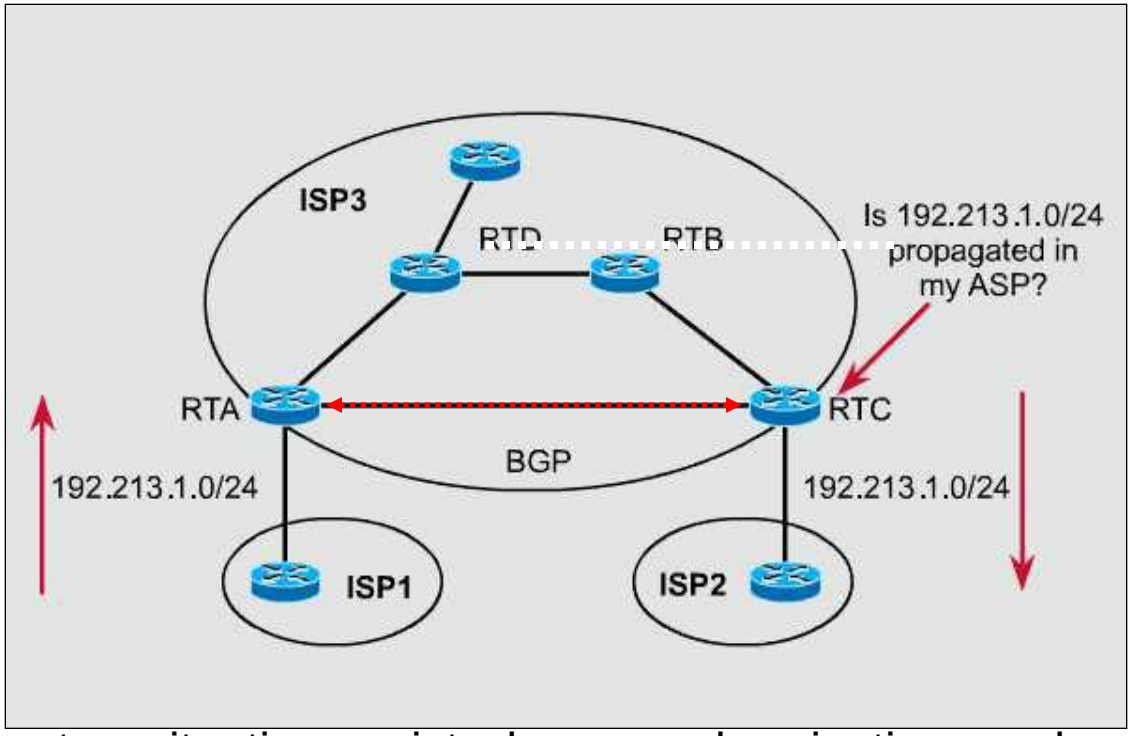
- The consequence of injecting BGP routes inside an AS is costly.
- Redistributing routes from BGP into the **IGP** will result in major overhead on the internal routers, which might not be equipped to handle that many routes.
- Besides, carrying all external routes inside an AS is not really necessary.

AS Synchronization



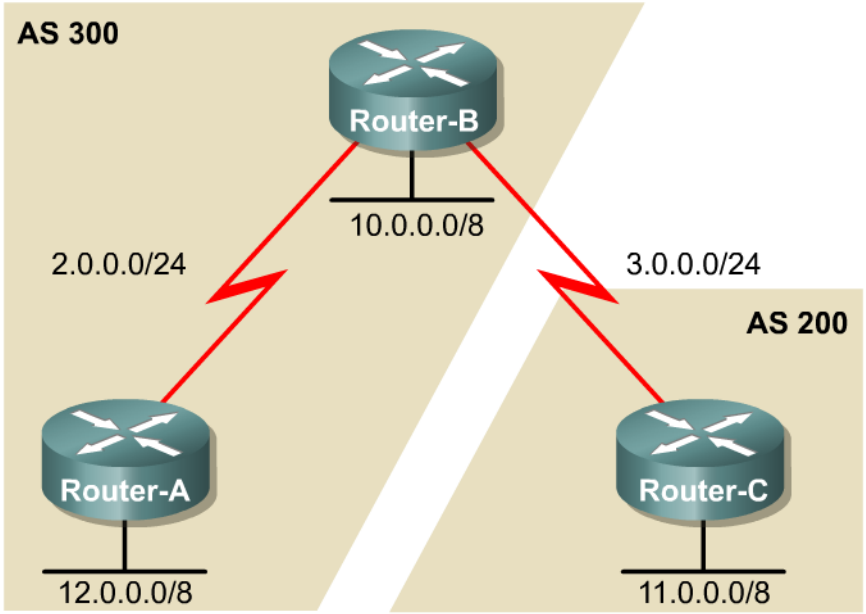
- The Cisco IOS offers an optional command called **no synchronization**.
- This command enables BGP to override the synchronization requirement, allowing the router to advertise routes learned via IBGP irrespective of an existence of an **IGP** route.

AS Synchronization



- In practice, two situations exist where synchronization can be safely turned off on border routers:
  - When all transit routers inside the AS are running fully meshed IBGP. Internal reachability is guaranteed because a route that is learned via EBGP on any of the border routers will automatically be passed on via IBGP to all other transit routers.
  - When the AS is not a transit AS.

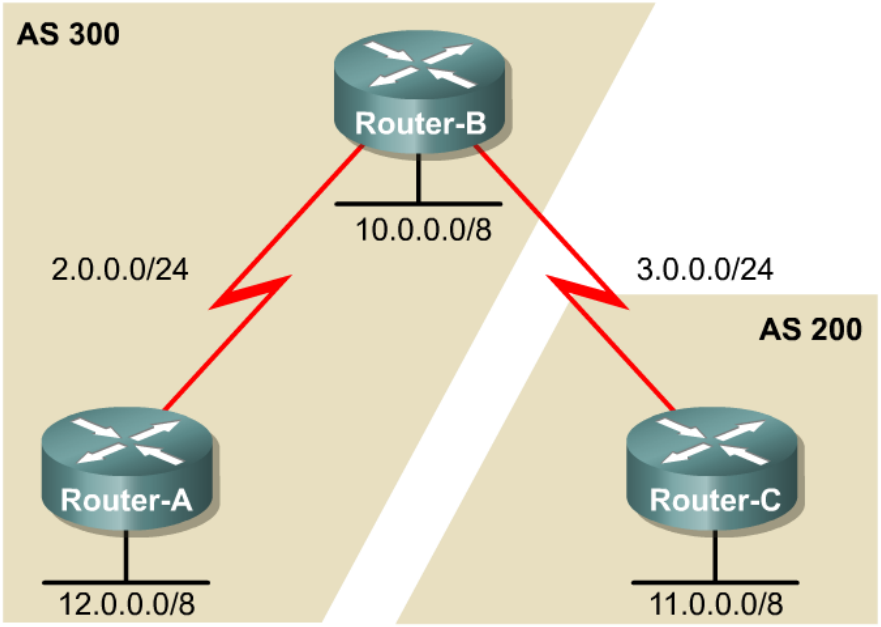
BGP Show Commands




```
Router-A#show ip bgp
BGP table version is 12, local router ID is 12.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - inte
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i10.0.0.0         2.0.0.1                  0    100      0 i
*>i11.0.0.0         2.0.0.1                  0    100      0 200 i
*> 12.0.0.0         0.0.0.0                  0         32768 i
Router-A#
```

```
Router-A#show ip bgp summary
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/

2.0.0.1         4   300     201     197       12    0    0 00:54:23
Router-A#
```

BGP Show Commands

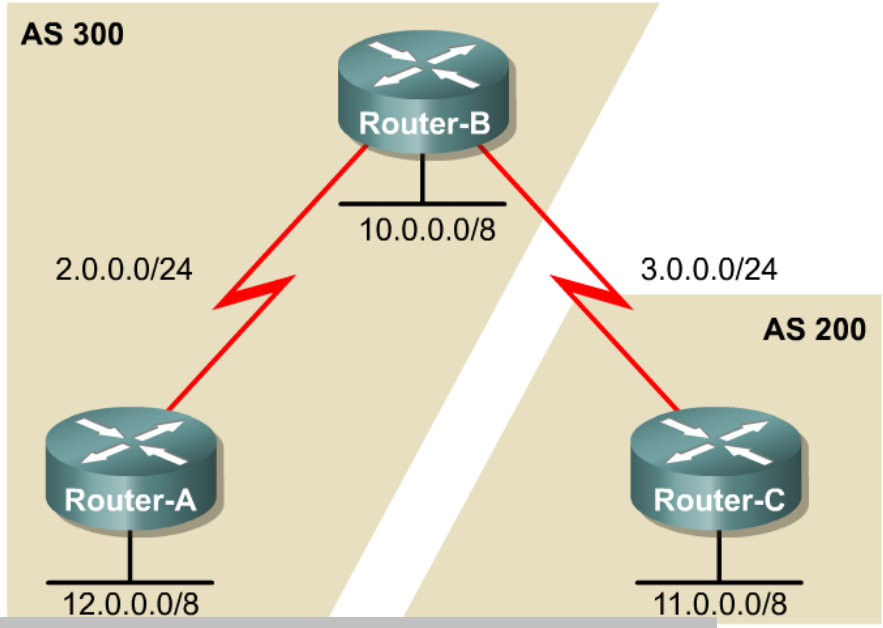



```
Router-B#show ip bgp
BGP table version is 12, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - inte
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.0.0.0         0.0.0.0                  0         32768 i
*> 11.0.0.0         3.0.0.2                  0             0 200 i
*>i12.0.0.0         2.0.0.2                  0    100      0 i
Router-B#
```

```
Router-B#show ip bgp summary
```

```
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/

2.0.0.2         4   300     202     206       12    0    0 00:59:51
3.0.0.2         4   200     172     175       12    0    0 02:17:05
Router-B#
```

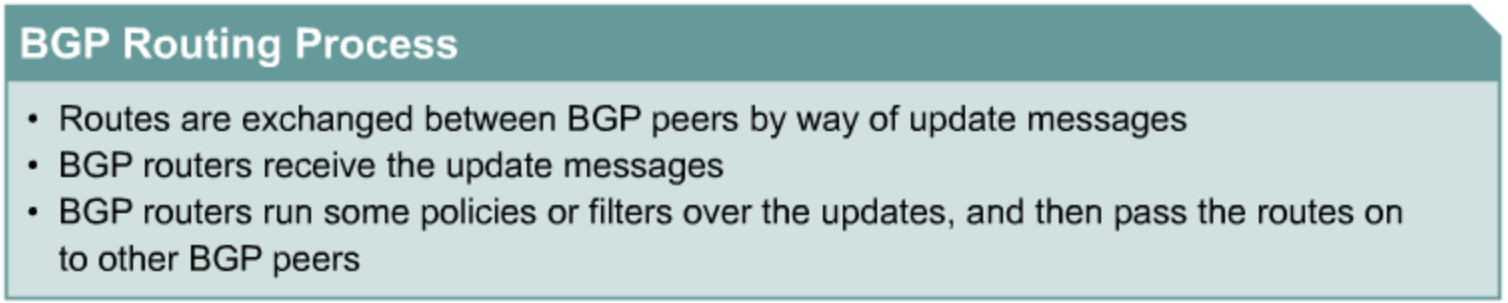
BGP Show Commands




```
Router-C#show ip bgp
BGP table version is 6, local router ID is 11.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - inte
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.0.0.0         3.0.0.1                  0             0 300 i
*> 11.0.0.0         0.0.0.0                  0         32768 i
*> 12.0.0.0         3.0.0.1                                0 300 i
Router-C#
```

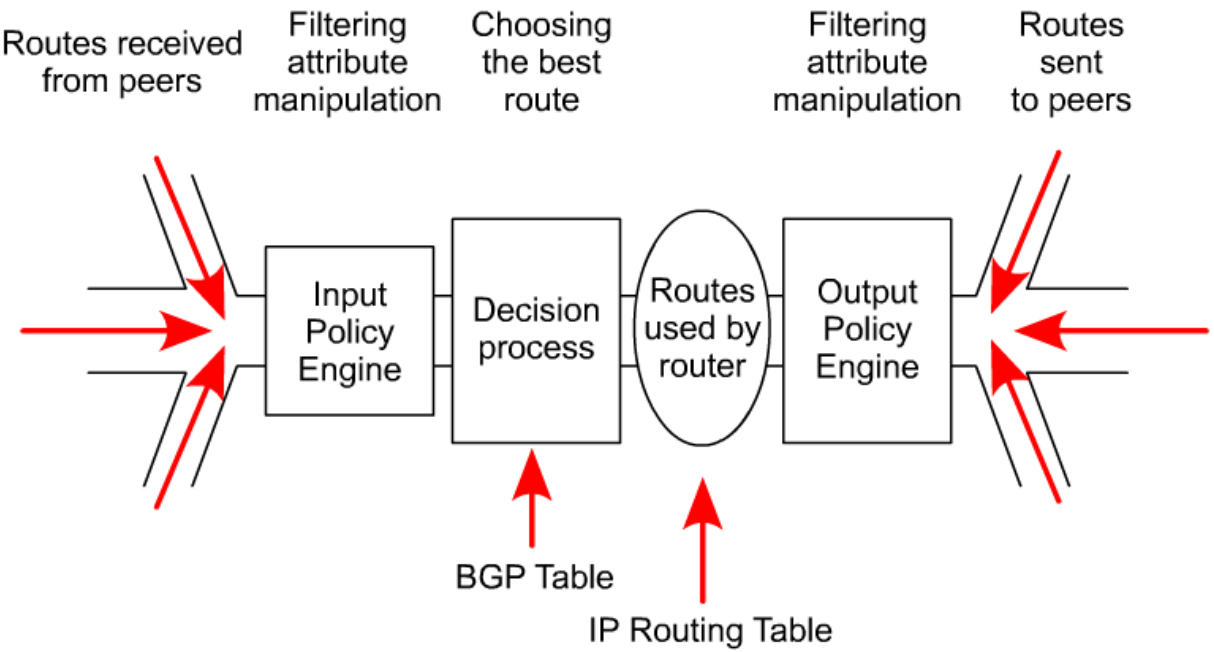
```
Router-C#show ip bgp summary
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/

3.0.0.1         4   300     178     176        6    0    0 02:20:15
Router-C#
```

BGP Routing Process



- The Cisco implementation of BGP keeps track of all BGP updates in a BGP table separate from the IP routing table.
- In case multiple routes to the same destination exist, BGP does not flood its peers with all those routes. Instead, BGP picks only the best route and sends it to the peers.
- In addition to passing along routes from peers, a BGP router may originate routing updates to advertise networks that belong to its own AS.
- Valid local routes originated in the system and the best routes learned from BGP peers are then installed in the IP routing table.
- The IP routing table is used for the final routing decision.

BGP Routing



- BGP is so flexible because it is a fairly simple protocol.
- Routes are exchanged between BGP peers via UPDATE messages.
- BGP routers receive the UPDATE messages, run some policies or filters over the updates, and then pass on the routes to other BGP peers.
- The Cisco implementation of BGP keeps track of all BGP updates in a BGP table separate from the IP routing table.

The Route Map Command


```
RTA(config)#route-map MYMAP permit 10
RTA(config-route-map)#match ip address 1
RTA(config-route-map)#set metric 5
RTA(config-route-map)#exit
RTA(config)#access-list 1 permit 1.1.1.0 0.0.0.255
```

- Router(config)#**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]

- BGP input and output policies are defined, generally, using route maps.
- Route maps are used with BGP to control and modify routing information and to define the conditions by which routes are redistributed between routing domains.
- Note that *map-tag* is a name that identifies the route map; the *sequence-number* indicates the position that an instance of the route map is to have in relation to other instances of the same route map.
- Instances are ordered sequentially, starting with the number 10 by default.

Applying a Route Map to BGP


```
RTA(config)#route-map MYMAP permit 10
RTA(config-route-map)#match ip address 1
RTA(config-route-map)#set metric 5
RTA(config-route-map)#exit
RTA(config)#access-list 1 permit 1.1.1.0 0.0.0.255

RTA(config)#route bgp 100
RTA(config-route)#neighbor 172.16.20.2 remote-as 300
RTA(config-route)#neighbor 172.16.20.2 route-map MYMAP out
```

- Access list 1 identifies all routes of the form 1.1.1.x.
- A routing update of the form 1.1.1.x will match the access list and will be propagated with a metric set to five (5).
- This is because of the **`permit`** keyword in the access list.
- A route map can be applied on the incoming, using the keyword **`in`**, or the outgoing, using the keyword **`out`**, BGP updates.
- The route map MYMAP is applied on the outgoing updates toward BGP neighbor 172.16.20.2.

STOP!

- Next Week, BGP Part 2:
  - BGP Attributes
  - The BGP Decision Process
  - BGP Route Filtering and Policy Routing
  - Redundancy, Symmetry, and Load Balancing
  - BGP Redistribution

- Let's stop here and go to the presentation:
  - Basic BGP Lab Configuration (PowerPoint)

Ch. 9 – BGP (Part 1)

Cabrillo College

CCNP 1 version 3.0

Rick Graziani

Cabrillo College